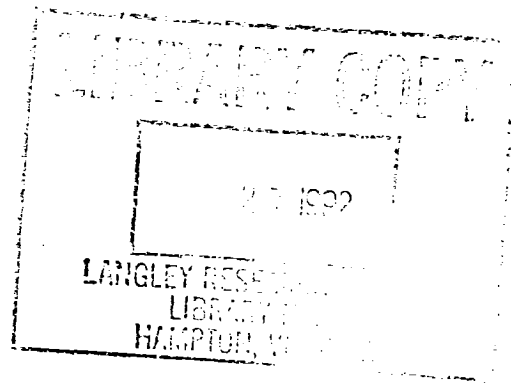


NASA Technical Memorandum 104168

**USER GUIDE FOR WIACX: A TRANSONIC WIND-TUNNEL
WALL INTERFERENCE ASSESSMENT AND CORRECTION
PROCEDURE FOR THE NTF**

Javier A. Garriz
Kara J. Haigler

January 1992



National Aeronautics and
Space Administration

Langley Research Center
Hampton, VA 23665-5225

FOR INFORMATION ONLY

NOT TO BE USED FOR OTHER PURPOSES

Introduction

WIACX is a three-dimensional transonic wind-tunnel Wall Interference Assessment and Correction (WIAC) procedure developed specifically for use in the National Transonic Facility (NTF) at NASA Langley Research Center. The codes comprising the procedure were developed from the CAP-TSD unsteady analysis code (described in references 1-4), and incorporate the WIAC ideas and procedures originally implemented in an earlier WIAC code (TUNCOR, described in references 5-7). The procedure is set up to perform two sequential calculations, as depicted in Figure 1, which illustrates the data flow in and major elements of a typical WIAC procedure. This document describes the procedures specifically designed for running WIACX using datasets from the NTF, including the preprocessors created to aid the user in constructing the test point-dependent input (the wall pressures used as outer boundary conditions in the "in-tunnel" simulation step of the calculations). The NTF wall pressure data preprocessor described herein is intended to be used in conjunction with several other existing utilities developed specifically for the NTF by Kemp. A sample case demonstrating the use of each of the elements in the procedure is included as a final section. It should be noted here that the grid used in this sample case is very coarse, resulting in poor resolution of the pressure distribution on the configuration's supercritical wing. It was chosen mainly for illustration purposes because it allowed running both WIACX codes (the "in-tunnel" and free-air halves) as a single job.

Although this document will deal exclusively with the application of WIACX to NTF datasets, the code has been applied to datasets from various wind-tunnel/model geometry combinations, ranging from single wings to a full wing-body-tail configuration. The reader is referred to references 8-11 for the details of such applications of WIACX.

General Description of Procedures

The procedures involved in preparing the input for the WIACX codes can be broadly classified into two (not entirely exclusive) groups. The first set of procedures performs the extraction and manipulation of the NTF wall pressure coefficient data (and associated aerodynamic coefficients) for each tunnel test point from the NTF data transmittal tapes. This first set of procedures produces the so-called "test-point-dependent" datafiles. The second (and considerably less

automated) group of procedures involves the setting up of input files containing the tunnel and model geometry description, the assignment of parameters used in the correction scheme embodied in WIACX and the appropriate array dimensions in the WIACX codes prior to execution. This second set of procedures produces datafiles which require at most only minor editing from test point to test point. The details pertaining to the function and use of each set of procedures will be discussed in the paragraphs that follow.

NTF Test Point-Dependent Data Processing Procedure:

The utilities and programs used to produce files containing the NTF wall pressure coefficient data and associated aerodynamic coefficients for each test point are shown in Figure 2 (all of the programs described in this section run under NOS 2.6). The NTFSIF and AERSIF programs are used to merge the NTF wall pressure and aerodynamic coefficient tapes and produce files in so-called "Standard Interface Format" (SIF). These two programs were developed by Kemp* along with several other routines used in PANCOR, a linear panel code WIAC procedure he developed. Either of the SIF files identified in Figure 2 can be additionally processed by using routines available in the Data Analysis Subsystem (DAS). Among the utilities contained in DAS are routines developed specifically for examining and editing the wall pressures along each of the nine rows of pressure taps in the NTF. Using DAS, the user can interactively view plots of the wall pressures on the individual rows, as well as smoothed cubic-spline fits of the experimental data. The utility allows the user to assign weighting factors to individual pressure orifices, as well as smoothing factors to individual rows, and promptly see the effect of such changes on the faired curves. The influence of bad pressure orifices on the faired wall pressures (which are to be used as outer boundary conditions) can thereby be controlled. A new SIF file containing the updated weighting and smoothing factors (if any) is produced, which is then ready for the next step in the procedure.

Assuming that the SIF file now contains both the NTF wall pressure coefficient data as well as the aerodynamic coefficients for a particular set of test points (the file identified on Figure 2 as the "final" SIF file), the next step involves the use of the PRECOR procedure to split the SIF file into separate, formatted

*Kemp's utilities are described in detail in the report "INTASYS: An Interference Assessment System for NTF — A User's Guide," which accompanies the distribution of his WIAC codes.

files containing the wall pressure and aerodynamic coefficient information for individual test points. This procedure is contained in a file called WIPROC and can be accessed by typing "GET,WIPROC/UN=310594C". The PRECOR procedure is then started by entering "-PRECOR,WIPROC,siffile", where 'siffile' is the name of the "final" SIF file that was the AERSIF program output. The program will then prompt the user to enter the reference area and reference length that were used in the normalization of the forces and moments for the particular model geometry (in units consistent with the rest of the tunnel and model description, as will be discussed in the next section). Next, the program will prompt the user to specify the test, run and point numbers of the particular test points for which the user desires wall pressure files for use in WIACX. The program applies the orifice weighting and row smoothing factors (contained in the SIF file) to the wall pressure coefficient data of each selected test point, and enriches the wall pressures by performing a cubic spline-fit in the cross-plane to interpolate for wall pressures on the upper and lower corners of the NTF test section. Figure 3 is an illustration of the portion of the NTF upper, lower, and side walls over which pressure taps are installed, and the relative position of the Pathfinder I model in the test section. The PRECOR procedure is set up to use only those pressures from the portions of the walls which contain more than one row of pressure taps. Therefore, none of the pressures measured past station 18.64 on any of the walls (see Figure 3) are used in the WIACX codes. PRECOR produces a pair of files per test point selected: a file containing the smoothed, weighted wall pressure and aerodynamic coefficients ready for use as one of the WIACX input files, and a file containing both the smoothed and experimental wall pressure coefficient data on each of the rows of wall taps (as well as the interpolated pressures on the upper and lower corners) for plotting. The files containing the WIACX input wall pressures are given names of the form "Tttrpp", where 'tt' are the last two digits of the test number, 'rr' are the last two digits of the run number, and 'pp' are the last two digits of the point number. The files containing the wall pressure plotting information are named in a similar fashion, with names of the form "Pttrpp". The contents of the second set of files can then be plotted (on a graphics terminal, of course) by accessing the plotting procedure PLTPROC by typing "GET,PLTPROC/UN=310594C" and then typing "-DI4014,PLTPROC,WALLPLT" after ensuring that the terminal is in Tektronix 4014 mode. The plot program will then prompt the user for the name of the file to be plotted, and proceed to plot the experimental, smoothed, and interpolated

(corner) wall pressure coefficients on each wall (one plot per wall).

The files produced by PRECOR reside as local files and must therefore be saved as permanent once they are deemed to contain satisfactorily smoothed and weighted wall pressures. At this point, the WIACX input wall pressure files must be transported to the Langley Cray-2, Voyager, where the remainder of the procedures for running WIACX are performed.

WIACX Code Correction Procedure:

As mentioned earlier, the codes comprising the WIACX procedure represent a merging of the WIAC procedures originally embodied in TUNCOR with the CAP-TSD code as a flow solver. One of the objectives in the development of WIACX was to leave as much of the input to (and output of) the original CAP-TSD code as unaltered as possible, so that any existing CAP-TSD input file description of a particular geometry would not have to be modified extensively for it to be usable in the WIACX procedure. As shown in Figure 1, the WIACX procedure is actually composed of two codes, "WIACXT" and "WIACXF". The WIACXT code seeks a solution of the transonic small-disturbance equation subject to the measured tunnel wall pressures as outer flow boundary conditions, with the additional requirement that specific measured flow properties be matched by the solution. In the case of a single wing, the total lift is constrained to be matched; in the case of a wing-tail, canard-wing, or wing-body-tail configuration (that is, a two-lifting-surface configuration), the total lift and pitching moment are constrained to be matched. This matching is done by varying the angle-of-attack of each lifting surface in proportion to the difference between the calculated and desired lift. The WIACXF code then solves the same governing equation subject to free-air boundary conditions (no perturbation in the far-field) with the requirement that the total lift calculated in free-air match that calculated in the "in-tunnel" simulation. The same scheme used to increment the angle(s)-of-attack in the WIACXT code is employed in WIACXF. In addition, WIACXF varies the free-stream Mach number in order to minimize the RMS difference between the Mach number distribution on the surface of the configuration in the tunnel and free-air calculations.

WIACX Code Input Data Processing Procedure:

In addition to the file containing the wall pressure and aerodynamic coefficient data described earlier, the input to the codes WIACXT and WIACXF consists of three files: one that contains the values of several parameters used in the angle-of-attack and Mach number incrementing schemes, and two standard CAP-TSD format input files (one describing the model configuration and computational boundaries in the tunnel and the other in free-air).

The two CAP-TSD input files containing the description of the configuration in the tunnel and in free-air are prepared in the standard CAP-TSD input format[†]. It is strongly recommended that the user become familiar with the preparation of standard CAP-TSD input files before using the WIACX procedure. While this document does not deal with the details involved in preparing such input files, a few suggestions on the construction of the "in-tunnel" and free-air grids are given in the next section. The only difference between these two files should be the boundaries of the computational domain in each; the grid used in the "in-tunnel" step will extend over that part of the wind-tunnel over which wall pressures were measured, while that used in the free-air step will extend several more characteristic lengths further in all directions. Once constructed, the user should only need to change the values of the configuration and lifting surface angles-of-attack (variables in namelist "config") as well as the Mach number (in namelist "flow") for running the WIACX codes at different test conditions.

The remaining file to be constructed contains two types of parameters: tunnel and model description-related parameters, and correction scheme-related parameters. The tunnel and model description parameters identify lifting surfaces as being either a wing or tail, and identify the type of boundary condition to be used on each wall of the tunnel in the "in-tunnel" step of the calculation. In the case of the NTF, the measured wall pressures would be used as Dirichlet boundary conditions on the side, lower and upper walls of the tunnel; however, the code is capable of using upper and lower wall shapes (slopes specified) as boundary conditions on those walls, providing that the appropriate "switches" are set in this file. The correction scheme-related parameters include such variables as the proportionality constant used in the calculation of the angle-of-attack increment,

[†]The document entitled "A Preliminary User's Manual for CAP-TSD (Version 1.0). A Program for Solving the Unsteady Transonic Small-Disturbance Equation for Aeroelastic Analysis of Realistic Aircraft Configurations" by J. T. Batina, D. A. Seidel, R. M. Bennett, S. R. Bland and R. W. Neely describes the input stream in detail. It accompanies the release of the CAP-TSD code to users.

the maximum allowable angle-of-attack increment, the initial Mach number increment, the incrementing and decrementing factors used in the calculation of the Mach number increment, and the maximum allowable absolute values of the lift error and Mach number increment at convergence. This file also contains the variable that determines if the user desires to have the wing and tail angles-of-attack driven independently in free-air (according to their respective lift mismatches), or instead vary a single "configuration" angle-of-attack to match the total wing, tail and body lift. Once constructed, this last file should not need to be edited at all for running the WIACX codes for several test points on a particular model in a given tunnel, unless the user decides to experiment with the correction scheme for particular test points. More detailed descriptions of each of these variables are given in the following section, which describes each of the codes individually.

WIACX Code Execution Procedure:

Having constructed the four input files, all that remains for the user is to dimension the codes WIACXT and WIACXF to accommodate the proper grid sizes (along with a few other correction scheme-related arrays described in the next section), ascertain the resources required (time and memory) to run WIACXT and WIACXF on Voyager, and submit the job to the appropriate queue. The additional arrays introduced to the CAP-TSD code to generate the WIACXT and WIACXF codes do not increase memory requirements substantially over that required by the original version of CAP-TSD. The time requirements are grid-size dependent, and are most accurately arrived at by actually running each code (WIACXT and WIACXF) a single iteration and examining the output to see how much time was consumed. The file "wiacx.sub" located in ~garriz/WIACX.demo1 directory is a typical submittal file which is submitted by entering "qsub wiacx.sub" while in the directory containing the input files and wiacx codes. This submittal file also takes care of renaming critical output files produced by each code (WIACXT and WIACXF), affixing a "t" or "f" to the file names to designate their origin (tunnel or free-air calculation).

WIACX Code Output Data Processing Procedure:

Both WIACXT and WIACXF programs produce several output files (almost identical to those produced by CAP-TSD), two pairs of which are of particular

interest. The output files "fort.21" and "fort.31" (both of which are produced by each WIACXT and WIACXF execution) are the only "standard" CAP-TSD output files that the user typically will examine to assess the validity of the run. The "fort.21" files (renamed "fort.21t" for the tunnel solution and "fort.21f" for the free-air solution) have not been modified by either of the WIACX codes, and contain the configuration surface pressures output in the normal CAP-TSD format. The "fort.31" files produced by the WIACX codes (also renamed as "fort.31t" and "fort.31f") have been modified to contain the iteration histories of several additional variables of interest. In addition to the regular contents of a CAP-TSD "fort.31" file, the "fort.31t" file contains the iteration histories of the wing and tail angles-of-attack, as well as the iteration histories of the wing and tail errors (mismatch from the desired values). The "fort.31f" file contains such information, as well as the iteration histories of the free-stream Mach number and the RMS deviation of the surface Mach number distribution from that in the tunnel calculation. The very last entries in the "fort.31f" file are the actual corrections to the angles-of-attack of the wing and tail, the Mach number correction, and the corrected conditions themselves, as will be shown in the sample case.

Program "capcp.f" plots the contents of the "fort.21" files, and "plcap.f" plots iteration histories of selected quantities in the "fort.31" files. Plotting the contents of the "fort.21" files allows the user to compare the calculated horizontal lifting surface pressures to any that may have been measured during the experiment, so that the fidelity of the solution in reproducing pertinent flow features can be assessed. Plotting the contents of the modified "fort.31" files allows the user to evaluate the degree of the convergence of both the tunnel and free-air solutions. The user quickly can determine the "goodness" of the lift and moment match by seeing if the wing and tail lift errors drop below the user-specified tolerance and if the wing and tail angles-of-attack reach steady-state values.

Each of these plot codes require: 1) examining and modifying, if necessary, parameter statements in the source code being used; and 2) setting up of an input control file which names files to be accessed and reads key variables into the program using namelist format. These plot codes must be executed on a computer with the LARCGOS (LUV0) plotting library and the LIBFTEK interface library for Tektronix interactive screen display capability. Interactive display of the plots, consequently, requires using a terminal or workstation capable of Tektronix mode viewing.

In addition to the pairs of "fort.21" and "fort.31" output files just described, the WIACXT and WIACXF codes together produce three files that the user will need to retain. The WIACXT code produces a file called "fort.24" which contains correction-scheme-related information that the WIACXF code uses, and a file called "fort.39" which, together with the "fort.40" file that the WIACXF code outputs, contains information that is used in approximating a buoyancy correction to the drag coefficient. The output files "fort.39" and "fort.40" become input files without further editing (together with the wall pressure and aerodynamic coefficient data file) to a post-processor for calculating the buoyancy correction and obtaining the aerodynamic coefficients renormalized to the corrected conditions. The details pertaining to the use of this post-processor are given in the sample case.

Detailed Description of Individual Code Input:

In this section, the inputs to each of the codes comprising the WIACX procedure are described in detail. In many cases it may be instructive to refer to the final section (the sample case) for further clarification.

Precor Input:

The PRECOR procedure (contained in procedure file WIPROC) calls a program called GNTCINA which reads in the "final" SIF file containing the NTF wall pressures and aerodynamic coefficient data for a set of test points, and applies the orifice weighting and row smoothing factors (also contained in the final SIF file) to the experimental data. Upon execution of the procedure, the first user prompt is:

INPUT REFERENCE AREA USED IN COEFFICIENT NORMALIZATION:
?

The value specified here should have units consistent with those used in the rest of the problem description, including the tunnel and free-air grid specifications contained in the standard CAP-TSD input files (English units, specifically, as will be discussed shortly). This also applies to the value specified in response to the next prompt, which is:

INPUT REFERENCE LENGTH USED IN CM NORMALIZATION:
?

The code then responds with the final prompt:

INPUT TEST NO., RUN NO., NO. OF POINTS AND POINT NO.S

\$IN NRUN= ,NTEST= ,MPOINT= ,NPOINT= , , \$

where

NRUN	The run number.
NTEST	The test number.
MPOINT	The number of test points selected: default 1, maximum 10.
NPOINT	The array of point numbers.

The code searches through the input SIF file and produces the wall pressure and aerodynamic coefficient data file as well as the wall pressure plot input file for each of the test points selected. Upon successful completion of the procedure, the code responds with

/DATA PREPROCESSOR siffile

Entering a "STATUS,F" will list the local files created during the interactive session, including the pair(s) of files just created. The first portion of the "Tttrpp" files contain a summary of the test point information, followed by the actual pressure measurements that are used as the outer boundary conditions in the WIACXT code. Several descriptors (such as "nx2", "nz2", "nx3", "ny3", etc.) which specify the number of rows of wall pressures on each wall and the number of taps on each row precede the actual wall pressures. The WIACXT code contains arrays whose dimensions depend on the values of these descriptors, but which do not have to be redimensioned when working with a specific wall pressure tap arrangement. The code is set up to accept the NTF wall pressure tap arrangement as depicted in Figure 3. The interested user is referred to the first portion of subroutine "datta" which contains detailed descriptions of each of these descriptors.

The code GNTCINA is set up to specify the NTF wall pressure tap locations in inches, with the origin as shown in Figure 3. This leads to the requirement that the remainder of the problem description (most especially the grids) be specified in units consistent with that choice.

As mentioned earlier, the wall pressure and aerodynamic coefficient data file(s) must be transported to Voyager for use with the WIACX codes.

WIACXT and WIACXF Input:

Preparing the file that describes the model configuration in the tunnel is the most time-consuming of all the input preparations that the user must undertake. However, once the user is familiar with the standard CAP-TSD input file format, following a few guidelines in the construction of this file will allow it to be easily modifiable so that it can be used to construct the input to the free-air calculation. While it is not a requirement that the "in-tunnel" grid be a proper subset of the free-air grid, such an arrangement would be very convenient, as illustrated in Figure 4. It is a requirement, however, that the portion of the grids over which the surface Mach number matching is done (the grid over the entire configuration surface) be common to both tunnel and free-air grids. This is due to the fact that the matching scheme involves taking the difference between the Mach numbers calculated at each point on the configuration surface for free-air and those calculated on the same points for the "in-tunnel" step. Having a common grid over the configuration surface would obviously be a natural consequence of making the tunnel domain a proper subset of the free-air domain. Also, as long as the user retains the coordinate reference center in both domains, the entries into namelist "trnsfrm" (describing the transformation region boundaries) will remain essentially the same in both the tunnel and free-air files. In the free-air file, one would simply replace the points specifying the tunnel upstream, downstream, and far spanwise boundaries with the appropriate free-air values, leaving the coordinates describing the lifting surface transformation regions unchanged. Furthermore, the value of the configuration total moment reference location (variable "xtm" in namelist "config") will be identical in both files, as will all the variables which specify a location on the configuration itself (variables "xrefp" and "xhm", for instance). The user should, however, ascertain that the value specified as the configuration total moment reference location in both files agrees with the experimentally specified balance center location. The indices specifying lifting surface extents in all three directions (such as "ihle", "ihle", "jhl", "jhr", "khu", "khl", etc.) would be the only other parameters that would need to be altered in the free-air file as a result of the grid extent being different. It should be clear, then, that a judicious choice of domains can considerably reduce the effort involved in constructing both input files, and that once a tunnel input file is constructed, it can be used as a "template" for constructing the free-air input file. Running several test points, however, does require editing both the tunnel and free-air files to specify the test

Mach number (variable "xmach" in namelist "flow") and configuration and lifting surface angles-of-attack (variables "alfrel0" and "alh0" respectively, in namelist "config"). The values of the test Mach number, wing and tail angles-of-attack, reference area and length should be checked against those values found at the top of the wall pressure and aerodynamic coefficient ("Tttrpp") file.

Having constructed the "in-tunnel" and free-air CAP-TSD input files, the user is ready to specify the parameters in the file containing the tunnel/model description and correction scheme-related parameters. These parameters are described below.

First, there are the tunnel and model-description parameters (all of which are specified in i10 format):

- | | |
|-------|--|
| nhs | The integer number of horizontal lifting surfaces in the configuration. This value should be equal to that specified for the CAP-TSD program variable "nhs" (specified in a parameter statement in both WIACXT and WIACXF), and the value of CAP-TSD input variable "nhrzs". |
| nwing | An integer array dimensioned nwing(nhs), identifying the horizontal lifting surface corresponding to the wing. If lifting surface number n is the wing, for example, then nwing(n) should be set equal to 1; otherwise it should be set equal to 0. There will be nhs rows of nwing values. The variable nwing appears in common block "wtd" in WIACXT and WIACXF and is automatically dimensioned to "nhs" once the value of "nhs" is specified in the appropriate parameter statement. |

ntail	A integer array dimensioned ntail(nhs), identifying the horizontal lifting surface corresponding to the tail. If lifting surface number n is the tail, for example, then ntail(n) should be set equal to 1; otherwise it should be set equal to 0. There will be nhs rows of ntail values. The variable ntail appears in common block "wtd" in WIACXT and WIACXF and is automatically dimensioned to "nhs" once the value of "nhs" is specified in the appropriate parameter statement.
ipnlw	An integer value of 1 specifying a lower wall shape instead of wall pressures (along with neulw =1).
ipnuw	An integer value of 1 specifying an upper wall shape instead of wall pressures (along with neuuw =1).
neufsw	An integer key selecting the type of boundary condition to be used by WIACXT on the wind-tunnel "far spanwise wall" , i.e., the wall beyond the wing tip. = 0 : Always set equal to 0 for the NTF (Dirichlet boundary condition).
neulw	An integer key selecting the type of boundary condition to be used by WIACXT on the wind-tunnel "lower wall", i.e., the wall beyond the wing lower surface. = 0 : Usually set equal to 0 for the NTF (Dirichlet boundary condition). = 1 : If a lower wall shape (slopes) is to be specified instead of wall pressures. The variable ipnlw (above) must also be set equal to 1 when this option is exercised.

neuuw

An integer key selecting the type of boundary condition to be used by WIACXT on the wind-tunnel "upper wall", i.e., the wall beyond the wing upper surface.

= 0 : Usually set equal to 0 for the NTF (Dirichlet boundary condition).

= 1 : If a upper wall shape (slopes) is to be specified instead of wall pressures. The variable ipnuw (above) must also be set equal to 1 when this option is exercised.

Note that if upper or lower wall shapes (in terms of wall slopes) are specified instead of wall pressures, the procedure as currently set up requires that such information be contained in the input SIF file at the data locations where the wall pressures would have been.

The next set of input variables are the correction-scheme-related parameters (all of which are specified in f10.5 format):

nsing

An integer key specifying the "single-alpha" switch.

= 1 : Calculate a single "configuration" angle-of-attack in free-air. This option retains the relative wing and tail angles-of-attack calculated for the tunnel simulation, and varies a single alpha value to match the total configuration lift (wing plus fuselage plus tail) which was calculated in the tunnel simulation.

= 2 : Updates independently the wing and tail angles-of-attack according to their respective lift errors, leading to separate angle-of-attack corrections for the wing and tail. In the case of a single-lifting-surface (no tail), it may be set to either value.

mchw An integer key indicating whether to include the wing of the configuration in the surface Mach number matching.
= 0 : Do not include the wing.
= 1 : Include the wing.

mcht An integer key indicating whether to include the tail of the configuration in the surface Mach number matching.
= 0 : Do not include the tail.
= 1 : Include the tail.

mchf An integer key indicating whether to include the fuselage of the configuration in the surface Mach number matching.
= 0 : Do not include the fuselage.
= 1 : Include the fuselage.

mdim The integer number of points on the configuration surface over which the Mach number matching will be done. In terms of mchw, mcht, mchf and CAP-TSD input variables, this value can be calculated as follows:

$$\begin{aligned}
mdim = & \\
& mchw * \\
& [2 * (jhr_w - jhl_w + 1) * (ihle_w - ihle_w + 1)] + \\
& mcht * [2 * (jhr_t - jhl_t + 1) * (ihle_t - ihle_t + 1)] + \\
& mchf * [2 * (jfr - jfl + 1) * (ifte - ifle + 1) + \\
& (ifte - ifle + 1) * (kft - kfb + 1)]
\end{aligned}$$

where the subscripts w and t refer to the wing and tail, respectively. The one-dimensional arrays "fun" and "funbas" in common block "smach" (which appears in subroutines "initial", "potent" and "erval3" in both WIACXT and WIACXF) should be dimensioned to mdim.

ca	The real value of the proportionality constant used in the angle-of-attack incrementing scheme (see reference 6).
xinca	The real value of the magnitude of the maximum allowable angle-of-attack increment (degrees).
erra	The real value of the maximum allowable absolute value of the lift error at convergence, where the lift error is (calculated lift - desired lift)/desired lift.
xincm0	The real value of the initial increment to the Mach number.
c1	The real value of the incrementing factor used in the Mach number optimization scheme (again, see reference 6). The value of c1 should be set greater than 1.
c2	The real value of the decrementing factor used in the Mach number optimization scheme (again, see reference 6). The value of c2 should be set less than 1.
errm	The real value of the maximum allowable absolute value of the Mach number increment at convergence (used in WIACXF).

At this point, at least four data input files will have been constructed (depending on the number of test points to be assessed):

1. The NTF wall pressure and aerodynamic coefficient data file.
2. The "in-tunnel" CAP-TSD input file (for WIACXT).
3. The free-air CAP-TSD input file (for WIACXF).
4. The file containing the tunnel/model, and correction scheme-related parameters.

Clearly, there will be as many of the first three input file types as there will be test points to be assessed, the only differences between each of the CAP-TSD "in-tunnel" or free-air input files being the flow conditions and configuration angles-of-attack. As mentioned earlier, the parameters contained in the fourth

input file are seldom changed after some initial numerical experimentation and can therefore be used for most, if not all, of the test points of a given test.

The codes WIACXT and WIACXF are set up to accept input file types one and four as long as the files bear specific filenames. In each directory containing the information for one test point (one of each of the four file types), the wall pressure file must be defined as a file called "fort.37", while that containing the tunnel/configuration and correction scheme parameters must be designated as "fort.38". The submittal file, however, must be edited to reflect the filenames chosen for the "in-tunnel" and free-air CAP-TSD input files, as will be seen in the sample case. The code WIACXT produces a file called "fort.24" in addition to the usual output files (including the modified "fort.31" file) produced by CAP-TSD. This is an unformatted file containing information pertaining to the "in-tunnel" solution (such as the final angles-of-attack and surface Mach number distribution) that will be needed by the WIACXF code. The file is renamed "fort.4" before being used by WIACXF, as shown in the sample submittal file in the next section. Should the user choose to run the WIACXT and WIACXF codes individually for purposes of debugging, the "fort.4" file should be present (in addition to the four files described above) when running WIACXF.

The final step prior to actual submittal of the job is setting the proper dimensions in WIACXT and WIACXF. The values of the grid sizes (which must be set exactly to the appropriate *nxt*, *nyt*, and *nzt* values) must be specified. Also, and very importantly, every occurrence of the parameter statement specifying the value of "nhs" must be checked to ascertain that it contains the same value as the input variable "nhrzs" in the "in-tunnel" and free-air input files. Finally, the dimensions of the one-dimensional arrays "fun" and "funbas" should be specified as described earlier. The source codes "wiacxt.f" and "wiacxf.f" are identical except in two respects. A variable called "n1t2fa" is set to 1 in "wiacxt.f", signalling the code to perform those correction-scheme processes related to the "in-tunnel" modelling (such as using the wall pressures as outer boundary conditions, etc.); in "wiacxf.f", the same variable is set to 2, which signals the code to perform those processes related to the free-air modelling (such as varying the free-stream Mach number, etc.). The only other difference between the codes is the grid dimension specification (the *nxt*, *nyt* and *nzt* values). It is suggested that the user check to see if these are indeed the only two differences between the codes to be used.

Plot Code Input:

The "fort.21" and "fort.31" output files produced by either an "in-tunnel" or a free-air calculation become input files to plot codes developed specifically for them. The "fort.31t" file contains information used to create the following plots:

1. Residual and number of supersonic points versus iteration number.
2. Wing lift error versus iteration number
(and tail lift error if a tail is present).
3. Wing angle-of-attack versus iteration number
(and tail angle-of-attack if a tail is present).
4. Configuration pitching moment coefficient versus iteration number.

Similarly, the "fort.31f" file contains information used to create the following plots:

1. Residual and number of supersonic points versus iteration number.
2. Free-stream Mach number and RMS deviation of surface mach number distributions from "in-tunnel" and free-air solutions versus iteration number.
3. Configuration lift coefficient and lift error versus iteration number.
4. Configuration pitching moment coefficient and pitching moment error versus iteration number (if $n_{sing} = 1$).
5. The difference between the free-air configuration angle-of-attack and the final "in-tunnel" wing angle-of-attack versus iteration number (if $n_{sing} = 1$).

In order to plot the "fort.31" output files, both an "in-tunnel" "ft.inp" file and a free-air "ff.inp" file are usually constructed. The format of both follows:

- 1) Name of the "fort.31" file from which to obtain data for plotting. The name may be a pathname or simply a local file name.
- 2) The INP1 namelist input needed to indicate the choice of plots to be produced.

The namelist "INP1" variables for iterative histories plotting include the following:

ihalf	Integer key specifying the number of iterative output steps to plot. = 0 : Every step (default). = 1 : Every other step.
jreq	Integer key specifying disposition of plotting results. = 0 : display on screen. = 1 : save the "nosplt" plot file for post-processing (default). = 2 : display on screen and save "nosplt" file.
nconf	The integer value determining the horizontal lifting surfaces of the model configuration. = 1 : wing (default). = 2 : wing-tail. = 5 : tail.
nsing	An integer key specifying the angle-of-attack mode recorded on the "fort.31" file. = 1 : "single-alpha" configuration mode (as in the "fort.38" file). = 2 : independent wing and tail angle-of-attack adjustment (default).
nlt2fa	An integer key specifying the type of run results to process. = 1 : "In-tunnel" (default). = 2 : Free-air.

Plotting the "fort.21" output files involves setting up an input control file for a run having a format as given below:

- 1) Name of the "fort.21" file. The name, as applies for any name in this file, may be either a pathname or a local file name.
- 2) The NML namelist input needed for the run.

- 3) Name of the experimental data file (if experimental data is available for use and has been specified by namelist NML variables "inexpw" and/or "inexpt"). If both wing and tail experimental data exist, the tail data follows that of the wing using the same format.
- 4) Name of the "fort.14" grid output file whenever the configuration includes a fuselage (specified by namelist NML variable "nconf").

Namelist NML variables for surface pressures plotting include the following:

inexpt	<p>The integer key specifying file access of tail experimental data to be plotted against calculated results.</p> <p>= 0 : omit experimental data (default).</p> <p>= 1 : include experimental data. This setting requires variable kactt=0.</p> <p>= -1 : read the experimental data, but do not plot it. Plot only the calculated results at the span stations at which the experimental data was taken. This setting requires variable kactt = 0.</p>
inexpw	<p>The integer key specifying file access of wing experimental data to be plotted against calculated results.</p> <p>= 0 : omit experimental data (default).</p> <p>= 1 : include experimental data. This setting requires variable kactw=0.</p> <p>= -1 : read the experimental data, but do not plot it. Plot only the calculated results at the span stations at which the experimental data was taken. This setting requires variable kactw= 0.</p>
itplt	<p>The integer iteration number at which the user desires to plot the computed pressures.</p>

jreq

The integer specifying disposition of results of the plotting.

= 0 : view on the screen.

= 1 : save on a file only (default) for later processing.

= 2 : view on the screen and save on a file.

kactt

The integer key specifying the type of y-locations specified for plotting tail surface pressures.

= 1 : plot pressures along eta grid lines whose indices are to be specified. This setting requires setting variable inextp to 0 and inputting variables nyosa and nysa.

= 0 : plot pressures at specified eta/semi-span y-locations (default). Usually these y-locations do not fall on grid lines. Input can be of two types: a) no experimental data mode requires input of variables nyos and yos (default), and b) experimental data input mode (set inextp=1) specifies inputting already non-dimensionalized eta/semi-span y-locations from an experimental data input file.

kactw

The integer key specifying the type of y-locations specified for plotting wing surface pressures.

= 1 : plot pressures along eta grid lines whose indices are to be specified. This setting requires setting variable inexpw to 0 and inputting variables nyosa and nysa.

= 0 : plot pressures at specified eta/semi-span y-locations (default). Usually these y-locations do not fall on grid lines. Input can be of two types: a) no experimental data mode requires input of variables nyos and yos (default), and b) experimental data input mode (set inexpw=1) specifies inputting already non-dimensionalized eta/semi-span y-locations from an experimental data input file.

nconf	<p>The integer value determining the model configuration.</p> <p>= 1 : wing (default).</p> <p>= 2 : wing-tail.</p> <p>= 3 : wing-fuselage.</p> <p>= 4 : wing-tail-fuselage.</p> <p>= 5 : tail.</p> <p>= 6 : tail-fuselage.</p> <p>= 7 : fuselage.</p>
nfyosa	<p>An integer variable specifying the number of eta grid lines along which longitudinal fuselage surface pressures are plotted. Variable nfysa must also be specified.</p>
nfysa	<p>An integer array dimensioned nfysa(nyt) specifying the nfyosa j-indices of the grid lines along which longitudinal fuselage surface pressures plotting is done. Variable nfyosa must be specified.</p>
nif	<p>The integer nvalif i-stations along the fuselage at which to plot the fuselage circumferential surface pressures. Variable nvalif must also be specified.</p>
nvalif	<p>An integer variable specifying the number of i-stations along the fuselage at which to plot circumferential surface pressures. Variable nif must be specified.</p>

nyos	<p>An integer array dimensioned nyos(nhs) indicating the number of output y-locations of wing and/or tail surface pressures.</p> <p>nyos(n=wing) requires inputting nyos-number of yos(nyt,n=wing) values. Variable kactw must also be set to 0.</p> <p>nyos(n=tail) requires inputting nyos-number of yos(nyt,n=tail) values. Variable kactt must also be set to 0.</p>
nyosa	<p>A integer array dimensioned nyosa(nhs) specifying the number of output eta grid lines (y-locations) for the wing and/or tail.</p> <p>The nyosa(n=wing) value specifies the number of wing stations. Variable array nysa(nyt,n=wing) must be input and variable kactw must be set to 1.</p> <p>The nyosa(n=tail) value specifies the number of tail stations. Variable array nysa(nyt,n=tail) must be input and variable kactt must be set to 1.</p>
nysa	<p>An integer array dimensioned nysa(nyt,nhs) specifying output eta grid lines to plot wing and/or tail surface pressures and for which no experimental data exists to be plotted.</p> <p>The nyosa(n=wing) number-of-stations are specified as nysa(nyt,n=wing) integer values. Variable kactw must be set to 0 for this option to be valid.</p> <p>The nyosa(n=tail) number-of-stations are specified as nysa(nyt,n=tail) integer values. Variable kactt must be set to 1 for this option to be valid.</p>
spant	<p>The real value of the model tail semi-span in inches (default = 1.0).</p>

spanw	The real value of the model wing semi-span in inches (default = 1.0).
yos	<p>A real array dimensioned yos(nyt,nhs) specifying y-locations for the output of wing and/or tail surface pressures plots and for which no experimental data will be plotted.</p> <p>The nyos(n=wing) number-of-stations are specified as yos(nyt,n=wing) real values. Variable kactw must be set to 0 for this option to be activated.</p> <p>The nyos(n=tail) number-of-stations are specified as yos(nyt,n=tail) real values. Variable kactt must be set to 0 for this option.</p>

The user must construct an input control file for plotting the “in-tunnel” results, and another for plotting the free-air results. As a result of the different grid extents in the tunnel and free-air calculations, the two files will differ (most likely in the values of the indices of the grid lines along which the user desires plots).

The file containing the experimental wing surface pressures must of course conform to a specific format. This format will be discussed in the sample case. Also, when plotting the computed free-air wing pressures against any experimental results, the experimental pressures must be renormalized by the new free-air Mach number for an accurate comparison between the computation and experiment. A short “renormalization” code is provided which uses the file containing the experimental wing pressures as input and outputs a file in the same format but with renormalized wing pressure coefficients. The use of this code will also be demonstrated in the next section.

Sample Case

In this section, use of the WIACX procedure is demonstrated on a test point from the Pathfinder I model NTF test program. The Pathfinder I is a generic transport model used for several research purposes in the NTF (see for example reference 12). The test point chosen is point number 398 from run 18, test 21. Once again it should be noted that the grid used in this sample case is very

coarse, resulting in poor resolution of the pressure distribution on the Pathfinder's supercritical wing. It was chosen mainly because it allowed running both WIACX codes (the "in-tunnel" and free-air halves) as a single job on the Langley Cray-2, Voyager. A more suitable wing grid distribution can be found in the directory `~garriz/WIACX.demo3`, in a file called "n121at", which describes a 20-degree swept semi-span NACA0012 wing in the NASA Ames High Reynolds Number Channel I. This particular directory also contains the input file describing the wing in free-air, as well as the "fort.37" and "fort.38" files, illustrating the application of the WIACX codes to a single-lifting-surface configuration.

At several points in the description of the various codes and their respective inputs and outputs, it may be helpful to refer to Figure 5, which schematically diagrams the relationships between the various files and programs comprising the entire procedure.

It should also be noted that with the exception of the files produced by the PRECOR code, all the files referenced in the following subsections can be found in one (and in some cases, all) of three directories on the Langley Cray-2, Voyager. These are:

- | | |
|-------------------------------------|---|
| 1. <code>~garriz/WIACX.demo1</code> | Contains all input and output files describing a complete WIACXT and WIACXF run of the Pathfinder I in the NTF, point 398, using the "double-alpha" option. |
| 2. <code>~garriz/WIACX.demo2</code> | Contains all input and output files describing a complete WIACXT and WIACXF run of the Pathfinder I in the NTF, point 398, using the "single-alpha" option. |
| 3. <code>~garriz/WIACX.demo3</code> | Contains all input and output files describing a complete WIACXT and WIACXF run of the NACA0012 semi-span swept wing in the NASA Ames High Reynolds number Channel I. |

Precor Code Run:

The wall pressure and aerodynamic coefficients for this and several other test points are contained in file SIFTARE which can be accessed (on the Cyber 860 machines) by typing "GET,SIFTARE/UN=310594C". Assuming that the DAS utilities have been used to examine the wall pressures contained in this file (particularly those of test point 398), the WIPROC procedure is then executed by typing "-PRECOR,WIPROC,SIFTARE", and the user responds as follows:

INPUT REFERENCE AREA USED IN COEFFICIENT NORMALIZATION:

?143.136

(the value of the reference area in square inches)

INPUT REFERENCE LENGTH USED IN CM NORMALIZATION:

?5.74

(the value of the reference length in inches)

INPUT TEST NO., RUN NO., NO. OF POINTS AND POINT NO.S

\$IN NRUN= ,NTEST= ,MPOINT= ,NPOINT= , , \$

\$IN NRUN=18,NTEST=21,MPOINT=1,NPOINT=398\$

The code then searches through file SIFTARE for test point 398, extracts the wall pressures and aerodynamic coefficients for the case, applies the orifice weighting and row smoothing factors to the wall pressures, and signals a successful execution with "/DATA PREPROCESSOR SIFTARE". As a result, the two files "T211898" and "P211898" should now reside as local files. The "P211898" file can be plotted using the PLTPROC procedure by typing "-DI4014,PLTPROC,WALLPLT", and responding to the code's prompt (with the terminal in 4014 mode):

ENTER DATA INPUT FILE IN SINGLE QUOTES:

? 'P211898'

The screen will then display the smoothed and experimental wall pressures on the sidewall, and then the lower and upper walls (the user should hit "return" to move on to the next plot). Figures 6a-6c show the plots produced by using the PLTPROC procedure on the "P211898" file. The numbers alongside each curve give the location (in inches) of the particular row of wall taps relative to the origin as shown in Figure 7. The curves with no corresponding experimental data points represent the interpolated wall pressures at the corners of the upper and lower walls. The faired curves represent the data as it will be used by WIACXT with

no further smoothing. If the user is dissatisfied with any aspect of the fairing, the DAS utilities must be used on the SIF file containing the test point(s) of interest to refair the data.

WIACX Code Run:

At this point, the "T211898" file is transported to a directory on Voyager via the "ftp" utility. In this case, the files comprising this sample test point reside in two directories, ~garriz/WIACX.demo1 and ~garriz/WIACX.demo2. Both directories contain the files necessary for running the WIACX codes on one test point as well as output files of interest, as will be discussed shortly. The files "pathf.tunnel" and "pathf.freeair" in both directories are the CAP-TSD input files set up to describe the Pathfinder I model (one-half of it, actually) in the NTF and in free-air, respectively. These sample files can be used as guides for constructing CAP-TSD input descriptions of other models in the NTF. Note that both input files specify all the locations, distances and lengths in inches, consistent with the wall pressure data locations specified in the "T211898" file.

The "fort.38" file contains the tunnel/model description and correction scheme-related parameters for this particular tunnel/model combination. The contents of the "fort.38" file in ~garriz/WIACX.demo1 are shown in Figure 8. The value of "nhs" is 2, since the model has both a wing and a tail. The next two entries identify which of the lifting surfaces is a wing, and which is a tail. The first row variable "nwing" identifies the wing as horizontal lifting surface number 1, and the second row variable "ntail" identifies the tail as horizontal lifting surface number 2. If the user intends to use the "single-alpha" option in the free-air calculation, the wing must precede the tail in the numbering of the horizontal lifting surfaces. Otherwise, the wing and tail can be numbered in any order. The next row of entries describes the boundary conditions to be used by WIACXT. As explained earlier, the zeroes indicate that wall pressures are to be used as Dirichlet boundary conditions on all walls. This particular directory (~garriz/WIACX.demo1) contains a run using a value of 2 for the variable "nsing", indicating that the wing and tail angles-of-attack are to be driven independently in free-air to match the respective "in-tunnel" lift values. The ~garriz/WIACX.demo2 directory contains the identical test point run with the "single-alpha" option ("nsing" = 1). The value of the variable mdim is arrived at by applying the expression given in the previous

section using the fuselage and lifting surface values found in either "pathf.tunnel" or "pathf.freeair" files.

The contents of the submittal file "wiacx.sub" is shown in Figure 9. The grids chosen for this particular description of the Pathfinder I result in the memory requirement shown on the third line of the file. Since the "wiacx.sub" file is set up to run the WIACXT and WIACXF codes in sequence, the memory allotment requested should obviously be based on that for the larger of the two jobs (the free-air calculation). The user may opt for submitting each "half" of the calculation as a separate job, especially while conducting code familiarization, debugging, or numerical experimentation. In this case, one must make sure that the file "fort.4" (renamed from the "in-tunnel" output file "fort.24") be present when running the free-air calculation, in addition to the usual four input files.

Once the job has executed, several output files will reside in the directory, only a few of which will typically be retained by the user. These are the remaining files in either of the directories mentioned earlier. The "fort.4" file should be retained for use in the event that the user should choose to re-run the free-air part of the calculation for any reason. The two "fort.21" files contain the configuration surface pressures, while the two "fort.31" files contain the iteration histories of several quantities of interest, as described earlier. The "fort.14t" and "fort.14f" files contain grid information that is used in plotting fuselage surface pressures. Finally, each of the files "fort.39" and "fort.40" become input to a code which calculates the approximate buoyancy correction to the drag coefficient. The use of each of these files in their respective post-processors is discussed in the next subsections.

Buoyancy Correction Post-Processor Run:

The output files "fort.39" and "fort.40", together with the input file "fort.37" constitute the input to the post-processor code "buoyancy.f", which is found in each of the above-mentioned sample directories. The only preparation that the user must make before running the code is to assign the exact values of three variables specified in a code parameter statement. These are:

nxtl	The "nxt" integer value (number of xi grid lines in the streamwise direction) used in the "in-tunnel" step of the calculation (in WIACXT). This value must be specified exactly.
nxtf	The "nxt" integer value (number of xi grid lines in the streamwise direction) used in the free-air step of the calculation (in WIACXF). This value must be specified exactly.
jhr	The integer value of "jhr" as specified in the CAP-TSD "in-tunnel" and free-air geometry input files (which is common to both files). Again, this value must be specified exactly.

Having specified the parameters, the user can interactively compile, load and execute the code "buoyancy". The output file "buoy.out" is produced, the contents of which are shown (for the $nsing = 1$ case contained in the directory `~garriz/WIACX.demo2`) in Figure 10. The output echoes the experimental flow conditions and aerodynamic coefficients for the particular point number, then lists the computed angle-of-attack correction (of the configuration if $nsing = 1$ and of the wing if $nsing = 2$). The approximate buoyancy correction to the drag coefficient is listed next, followed by the correction to the drag coefficient resulting from the angle-of-attack correction. The ratio of the experimental and corrected dynamic pressures is given, along with the corrected aerodynamic coefficients (that is, the experimental lift and pitching moment coefficients renormalized to the corrected conditions, and the drag coefficient corrected to account for the buoyancy and angle-of-attack corrections and renormalized to the corrected conditions as well).

Plotting Post-Processor Runs:

The final entries to the "fort.31f" file are the angle-of-attack and Mach number corrections as well as the corrected conditions themselves. While these are the "final results" that the user seeks, it is always prudent to plot the results of the

"fort.21" and "fort.31" files. All the files necessary for producing these plots can be found in the ~garriz/WIACX.demol directory.

The iteration histories plot program, "plcap.f", processes and plots the "fort.31" file from a WIACXT or WIACXF code execution. The sample input control files used in this document may be accessed as "fl.inp" for the "in-tunnel" run and "ff.inp" for the free-air run. The contents of these plot input control files are shown in Figures 11 and 12. An example of each of the different type plots resulting from the specifications made in these sample input files is shown in Figures 13-21. A single code parameter statement, mxnres=5000, defines the maximum number of iterative output steps in a WIACXT or WIACXF code run. For a particular run this parameter must be set such that "mxnres"-of-the-plcap.f-code is greater than or equal to the "nstep"-of-the-fort.31-file.

The "fl.inp" file has been set up to access a local file, "fort.31t", after which the INP1 namelist input specifies an "in-tunnel" run to generate plots using data for the horizontal lifting surfaces, a wing and a tail. The namelist input also selects the double or independent alpha calculation of results format of the WIACXT output, although this setting is default and the only option for an "in-tunnel" run. For this run the "nstep"-of-the-calculation = 3000 is well within the currently set code parameter "mxnres" limit. After execution of the "plcap.f" code, the resulting "nosplt" file can be interactively viewed or printed.

In the "ff.inp" file, the variable "nlt2fa" set to 2 specifies a free-air run, and "nsing" set to 1 signals a single-alpha configuration mode run. Ordinarily the values for the remaining variables are unchanged from the other part of a complete set of runs. If the nstep-of-the-calculation is greater than the currently set value of 5000, the program will stop for resetting and re-compiling before code execution can be completed.

Plotting the configuration surface pressures contained in the "fort.21t" file will be demonstrated next. The input control file containing the file names to be accessed and the variables describing the desired configuration surface plots is named "cpinp.wtf", and is displayed in Figure 22. Notice that the "kactw" value combined with the "inexpw" value indicates that experimental wing pressures are to be read in from the "excp398" file and that the experimental pressure stations define the locations at which both calculated and experimental data will be plotted. The "nyosa" value indicates that surface pressures along one tail station ($j = 6$) are to be plotted, while the "nfyosa" value indicates that those along two longitudinal

fuselage stations ($j = 2$ and 3) are to be plotted. The "nvalif" value indicates that two stations ($i = 80$ and 120) along the fuselage have been selected for plotting the surface pressures along the fuselage circumference.

The "excp398" file contains the experimental surface pressures. The first portion of this file is shown in Figure 23. The "mexp" value is the nominal tunnel Mach number, while the "mcor" value corresponds to the corrected Mach number (listed as one of the final entries in the "fort.31f" file). The "ns" value corresponds to the number of stations along the wing at which pressures were measured. At each of these stations, the value of eta (nondimensional location along the semi-span) is specified along with the number of pressure coefficients specified on the upper ("ncpu") and lower ("ncpl") surfaces. The streamwise location of the pressure coefficients is specified along a nondimensional chord, as shown in Figure 23, for the first of the six measuring stations along the Pathfinder I wing. Any experimental tail surface pressures must be specified in the same format as the wing pressures, beginning immediately after all the wing values are listed. Hence, one would again specify the test, run and point numbers, as well as the tunnel and corrected Mach numbers before specifying the tail pressure values at each experimental tail eta station.

As mentioned earlier, the contents of this experimental pressure file must be renormalized by the "mcor" value before plotting them against the free-air surface pressures (in "fort.21f"). This can be accomplished by using the code "renorm.f". The code is set up to accept up to ten experimental span stations, with no more than thirty upper and lower surface pressure coefficients specified along each station (limits which can be changed by altering dimension statements in the source code). The user must then copy the "excp398" file into one called "cpexp", and interactively compile and run the code to produce a file called "cpfree" containing the renormalized pressure coefficients. The "cpfree" file must then be specified (in place of the "excp398" file) when plotting the free-air results.

Having constructed the "cpinp.wtl" and "excp398" files, the user must specify the "nxt", "nyt", and "nzt" values in a code parameter statement in the plot code "capecp.f". These values must be set equal or greater than the grid values used in the WIACXT code execution run. Lastly, the user should check that the value of "nhs" in another parameter statement is at least as large as the actual value in the particular WIACXT run. The user is then ready to compile and run the code on a machine on which the LARCGOS and LIBFTEK libraries reside. The plots

resulting from the choices indicated in "cpinp.wtf" are shown in Figures 24–34. Figures 24–29 show the computed versus experimental wing pressures. Figure 30 shows the computed tail pressure along the $j = 6$ grid line. Figures 31 and 32 are plots of the longitudinal upper and lower fuselage surface pressures along the $j = 2$ and $j = 3$ grid lines, respectively. Finally, Figures 33 and 34 are plots of the circumferential fuselage surface pressures (plotted against the nondimensional arc length along the fuselage circumference) at the $i = 80$ and $i = 120$ grid stations, respectively.

Plotting the contents of the free-air "fort.21f" file proceeds in the same fashion as described above for the "in-tunnel" results. In the plot input control file, the values specifying the indices of the wing, tail and fuselage grid lines along which plots are desired will be different (because the free-air grid is numbered differently), as will the names of the ".21", ".14", and experimental pressure files. Finally, the plot code parameter settings may need to be reset large enough to handle the free-air "nxt", "nyt" and "nzt" grid values.

Summary

The three-dimensional transonic code WIACX has been developed specifically for performing wind-tunnel Wall Interference Assessment and Correction calculations on data from the National Transonic Facility at NASA Langley Research Center. Using NTF wall pressures (appropriately smoothed and faired by preprocessors in the WIACX procedure) and measured force and pitching moment coefficients, the code calculates global corrections to configuration angles-of-attack and free-stream Mach number. Additionally, post-processors available in the procedure allow the calculation of the "corrected" force and moment coefficients (including estimated tunnel buoyancy corrections). Plotting the configuration surface pressures (computed and experimental) as well as the iteration histories of several quantities of interest is also available. These capabilities are demonstrated through a sample calculation performed on a point taken from the Pathfinder I test program.

Acknowledgments

NASA Langley Research Center sponsored this work under Contract NAS1-18585. The authors would also like to acknowledge the assistance of

Dr. Perry A. Newman and William B. Kemp, Jr. in the development of many of the programs comprising the WIACX procedure.

References

1. Batina, J. T.: "An Efficient Algorithm for Solution of the Unsteady Transonic Small-Disturbance Equation." AIAA Paper 87-0109, January 1987. (Also available as NASA TM-89014, December 1986.)
2. Batina, J. T.; Seidel, D. A.; Bland, S. R.; and Bennett, R. M.: "Unsteady Transonic Flow Calculations for Realistic Aircraft Configurations." *Journal of Aircraft*, Vol. 26, January 1989, pp. 21-28. (Also available as NASA TM-89120, March 1987.)
3. Batina, J. T.: "Unsteady Transonic Algorithm Improvements for Realistic Aircraft Configurations." AIAA Paper 88-0105, January 1988.
4. Batina, J. T.: "Unsteady Transonic Small-Disturbance Theory Including Entropy and Vorticity Effects." NASA TM-100568, March 1988.
5. Rizk, M. H.; Hafez, M.; Murman, E. M.; and Lovell, D.: "Transonic Wind Tunnel Wall Interference Corrections for Three-Dimensional Models." AIAA Paper 82-0588, March 1982.
6. Rizk, M. H.; and Murman, E. M.: "Wind Tunnel Wall Interference Corrections for Aircraft Models in the Transonic Regime." *Journal of Aircraft*, Vol. 21, January 1984, pp. 54-61.
7. Rizk, M. H.: "Improvements in Code TUNCOR for Calculating Wall Interference Corrections in the Transonic Regime." AEDC-TR-86-6, 1986.
8. Newman, P. A.; Kemp, W. B., Jr.; and Garriz, J. A.: "Emerging Technology for Transonic Wind-Tunnel Wall Interference Assessment and Corrections." SAE Technical Paper No. 881454, October 1988.
9. Newman, P. A.; Kemp, W. B., Jr.; and Garriz, J. A.: "Wall Interference Assessment and Corrections." NASA CP-3020, April 1988, Vol. 1, Part 2, pp. 817-851.
10. Garriz, J. A.; Newman, P. A.; Vatsa, V. N.; Haigler, K. J.; and Burdges, K. P.: "Evaluation of Transonic Wall Interference Assessment and Corrections for Semi-Span Wing Data." AIAA Paper 90-1433, June 1990.
11. Green, L.; Zhang, Q.; Garriz, J.; Wang, S.; Vatsa, V.; Haigler, K.; and Newman, P.: "NASA/CAE Wind Tunnel Interference Cooperative Program—Status and Sample Results, January 1991." Paper W-1 at the International Conference on Adaptive Wall Wind Tunnel Research and Wall Interference Correction, Xian, Shaanxi, China, June 1991 (Sample results from

the NASA-CAE cooperative agreement on Wall Interference Assessment and Correction, held in October 1990 and June 1991).

12. McKinney, L. W.; Bruce, W. E., Jr.; and Gloss, B. B.: "National Transonic Facility Status." NASA CP-3020, April 1988, Vol. II, pp. 1-39.

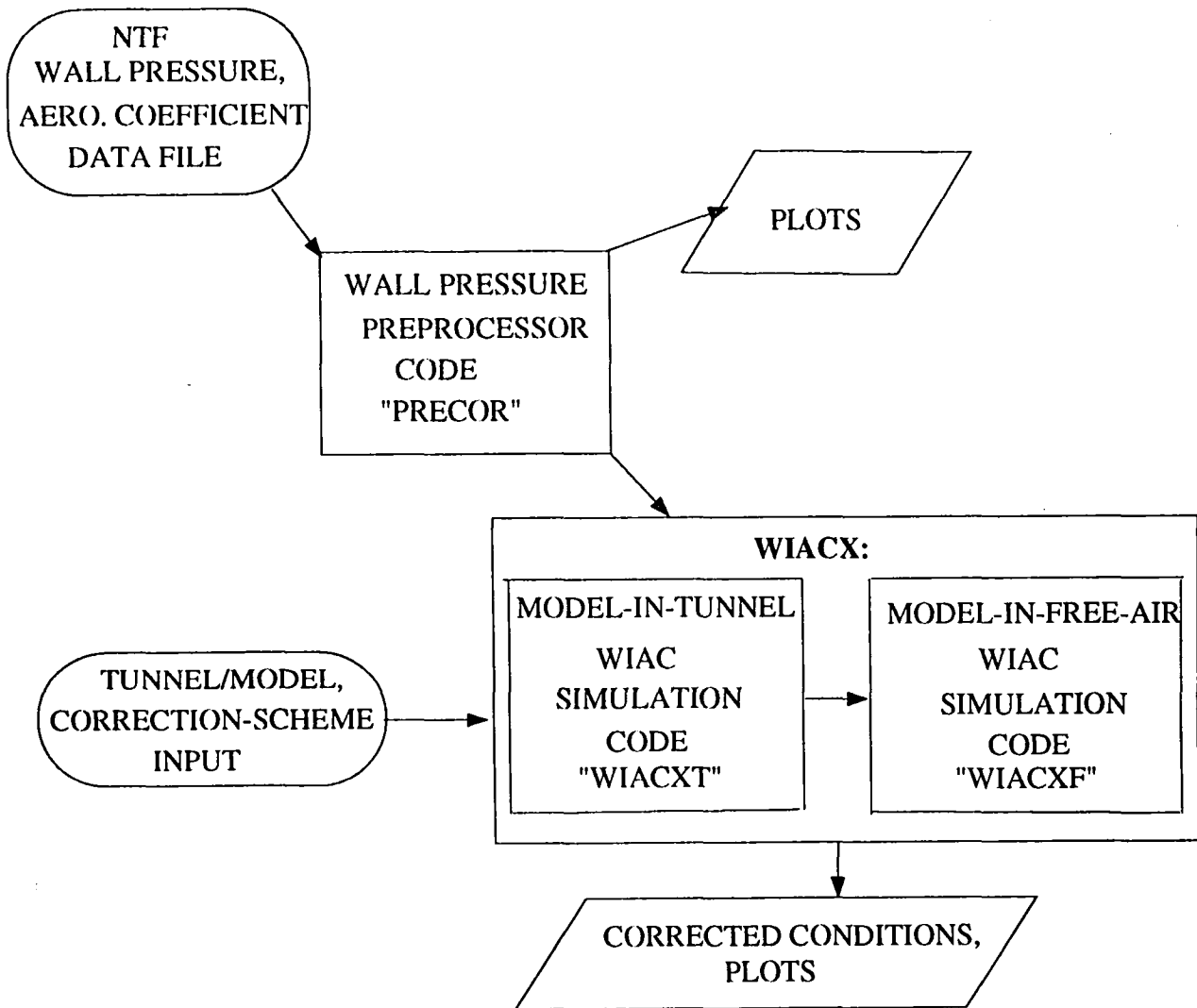


Figure 1. Elements of WIACX Procedure

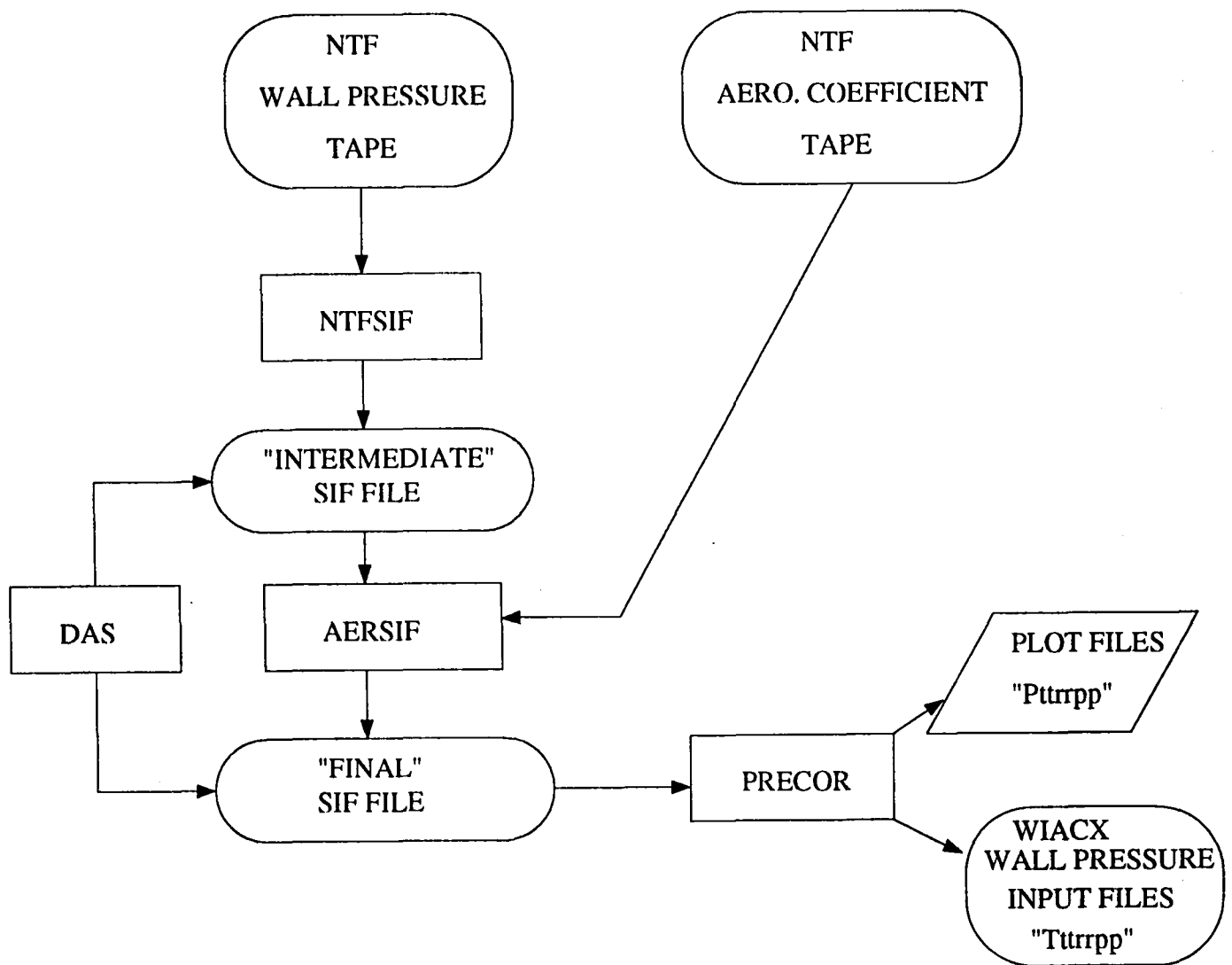


Figure 2. Data Stream in NTF Wall Pressure File Input Preparation.

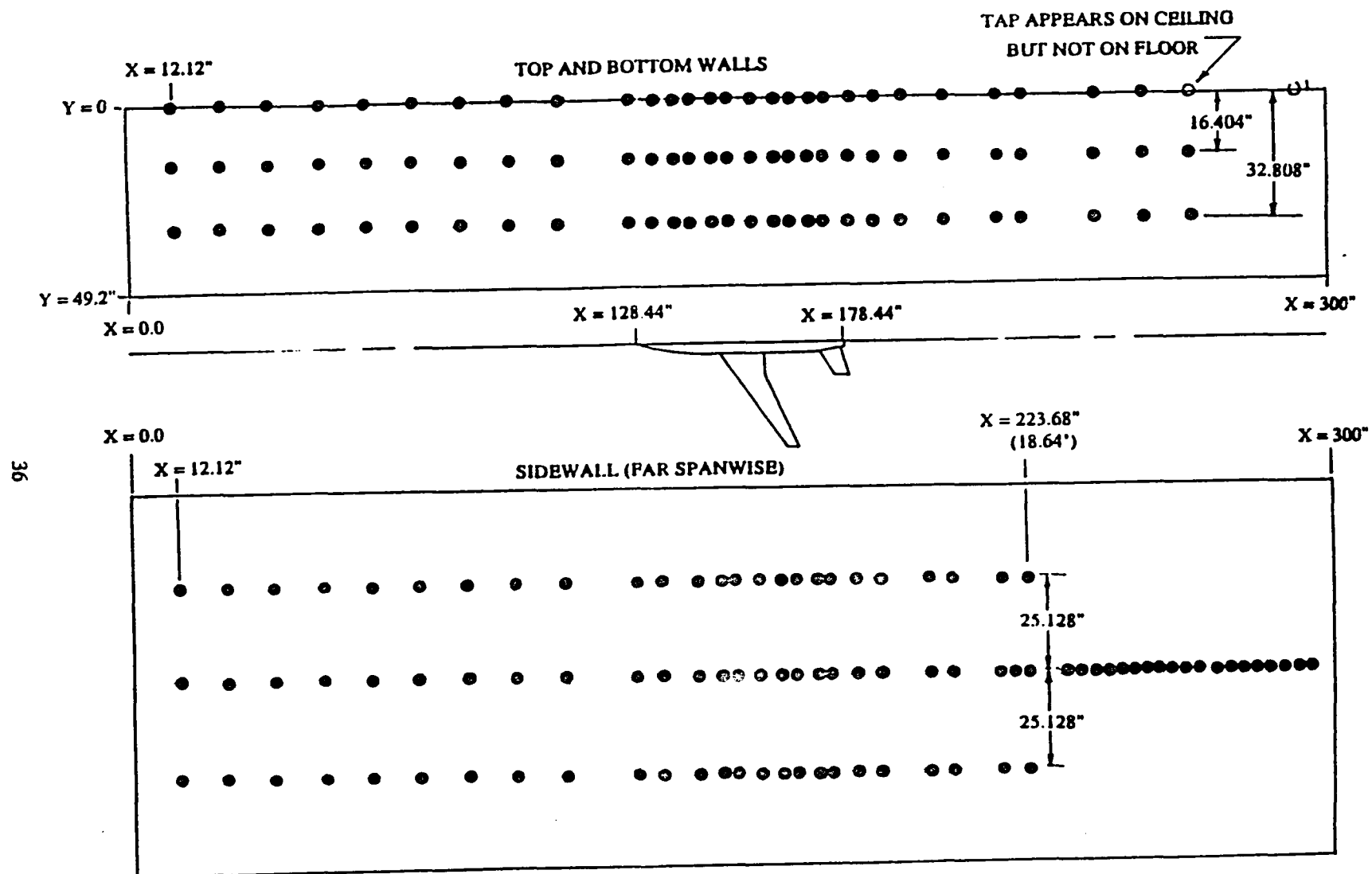


Figure 3. NTF Wall Pressure Tap Arrangement.

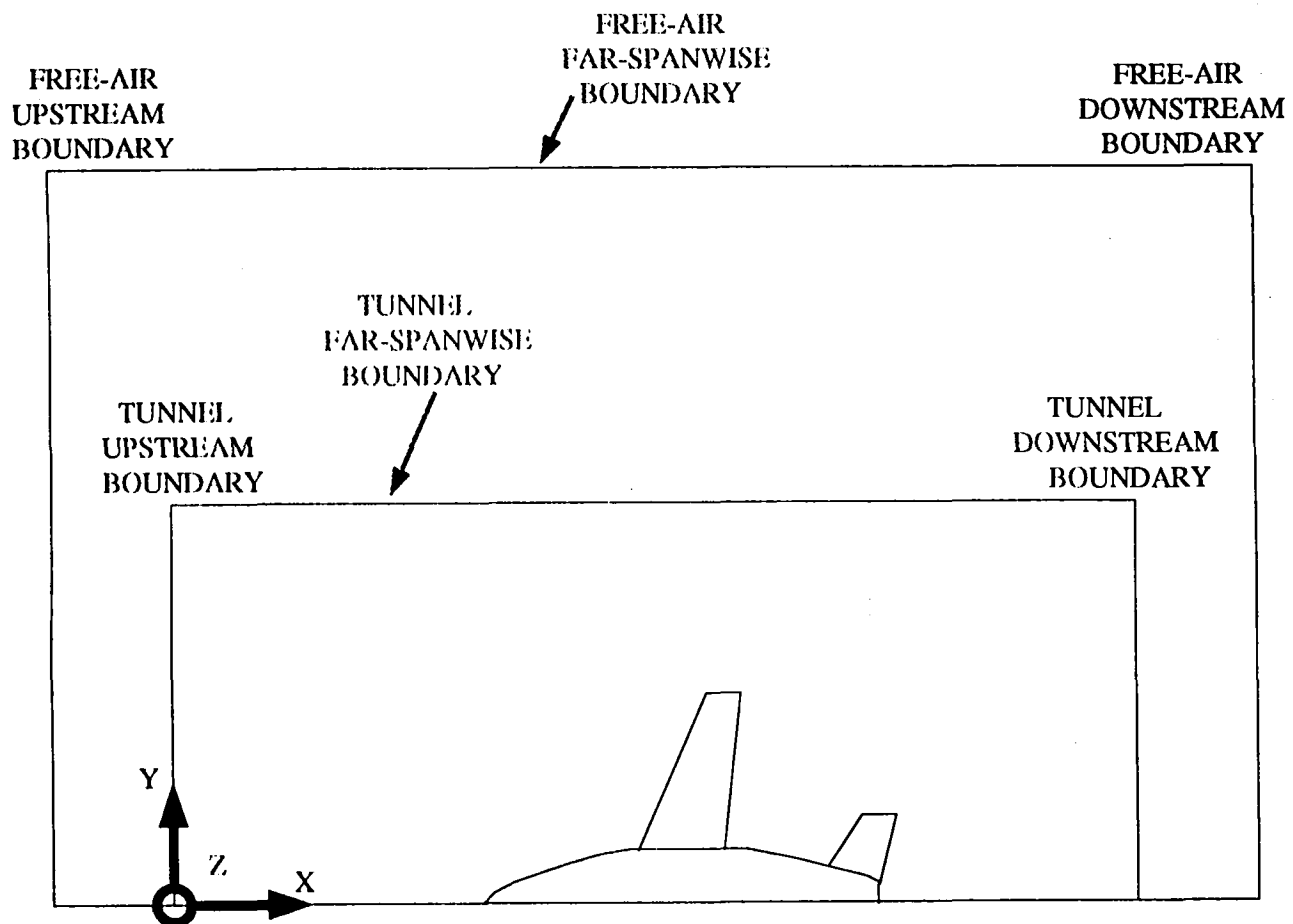


Figure 4. Tunnel and Free-air Grid Boundaries.

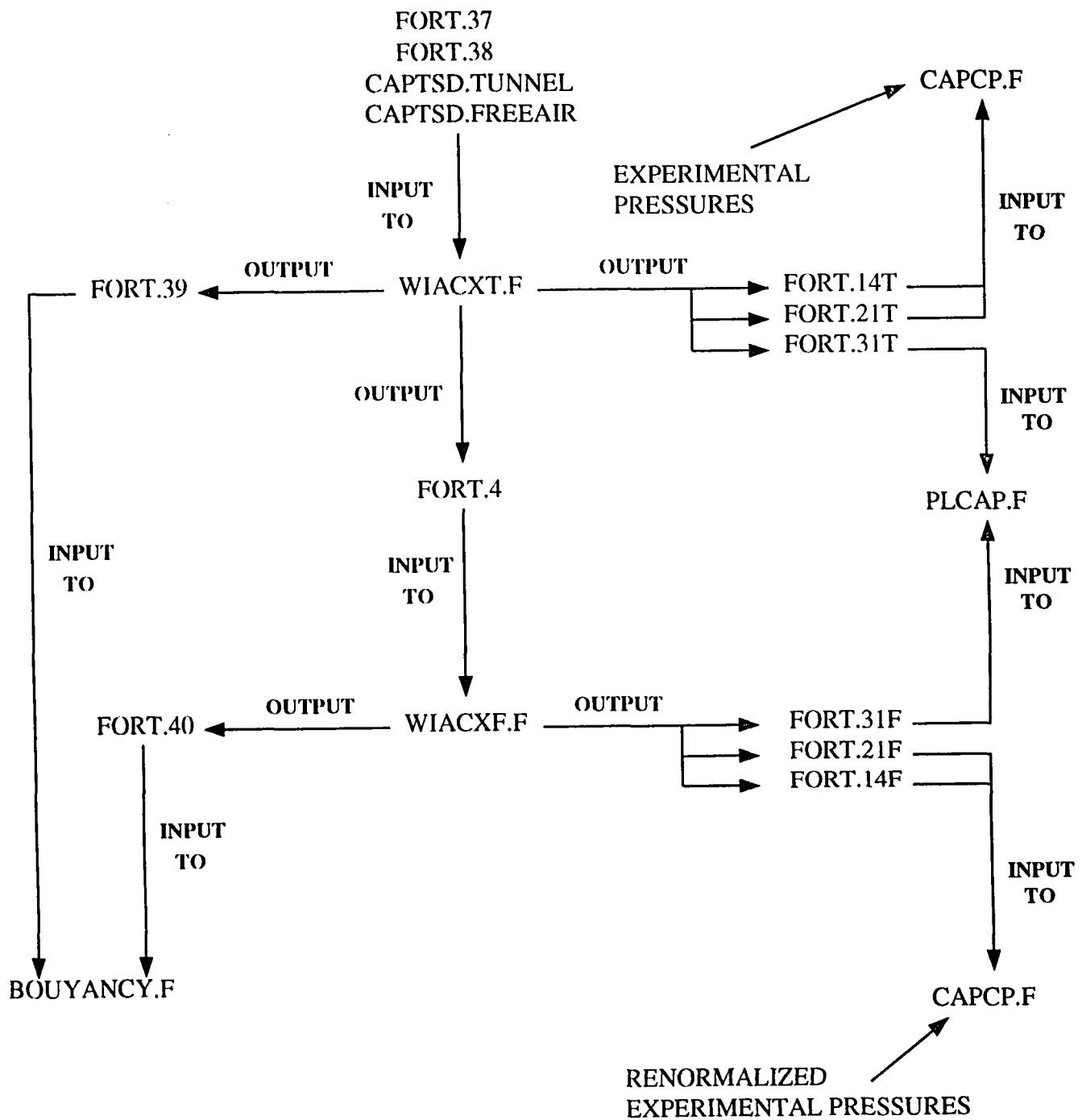
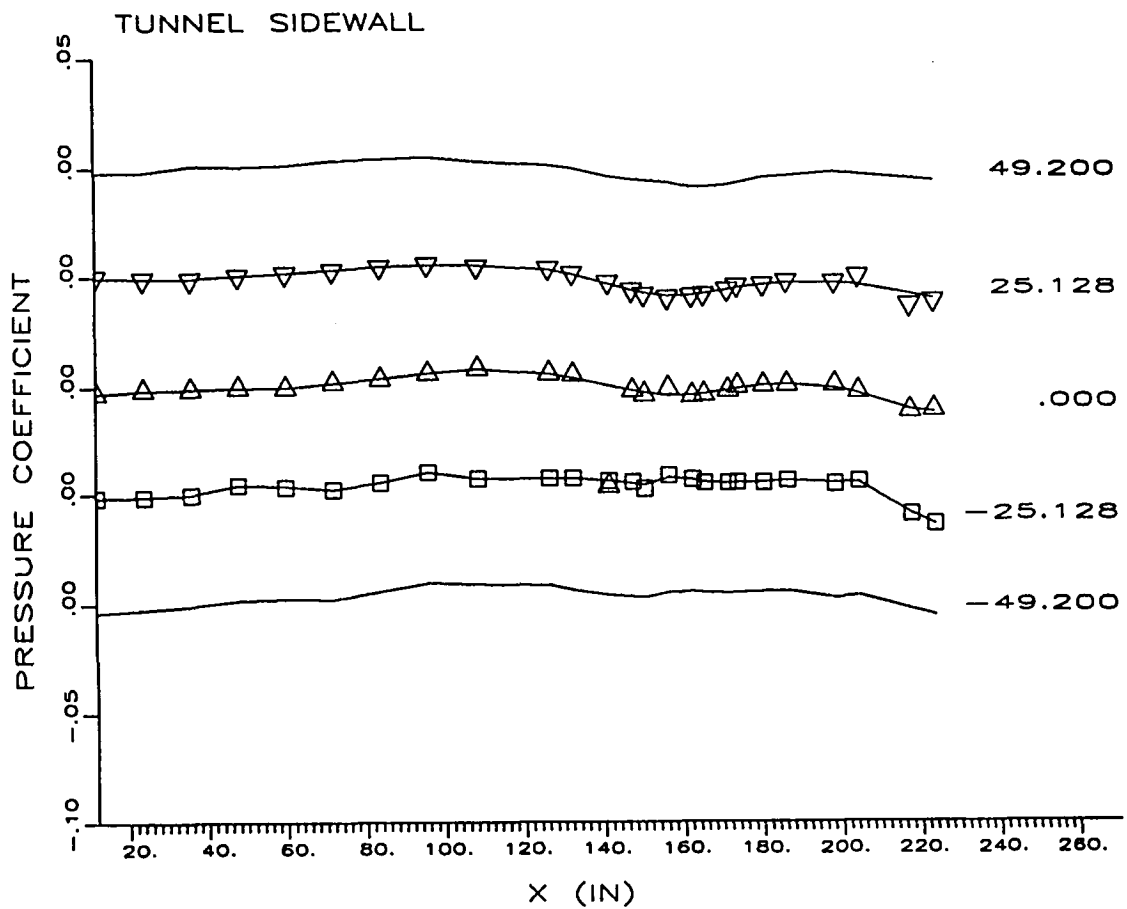


Figure 5. Input and output file relationships in WIACX procedure.



NTF TEST POINT 398 MACH= .7998 ALPHA= 2.2205

Figure 6 (a). Plot of Experimental (Symbols), Smoothed (Solid Lines), and Interpolated (Solid Lines With No Corresponding Symbols) Sidewall Pressures.

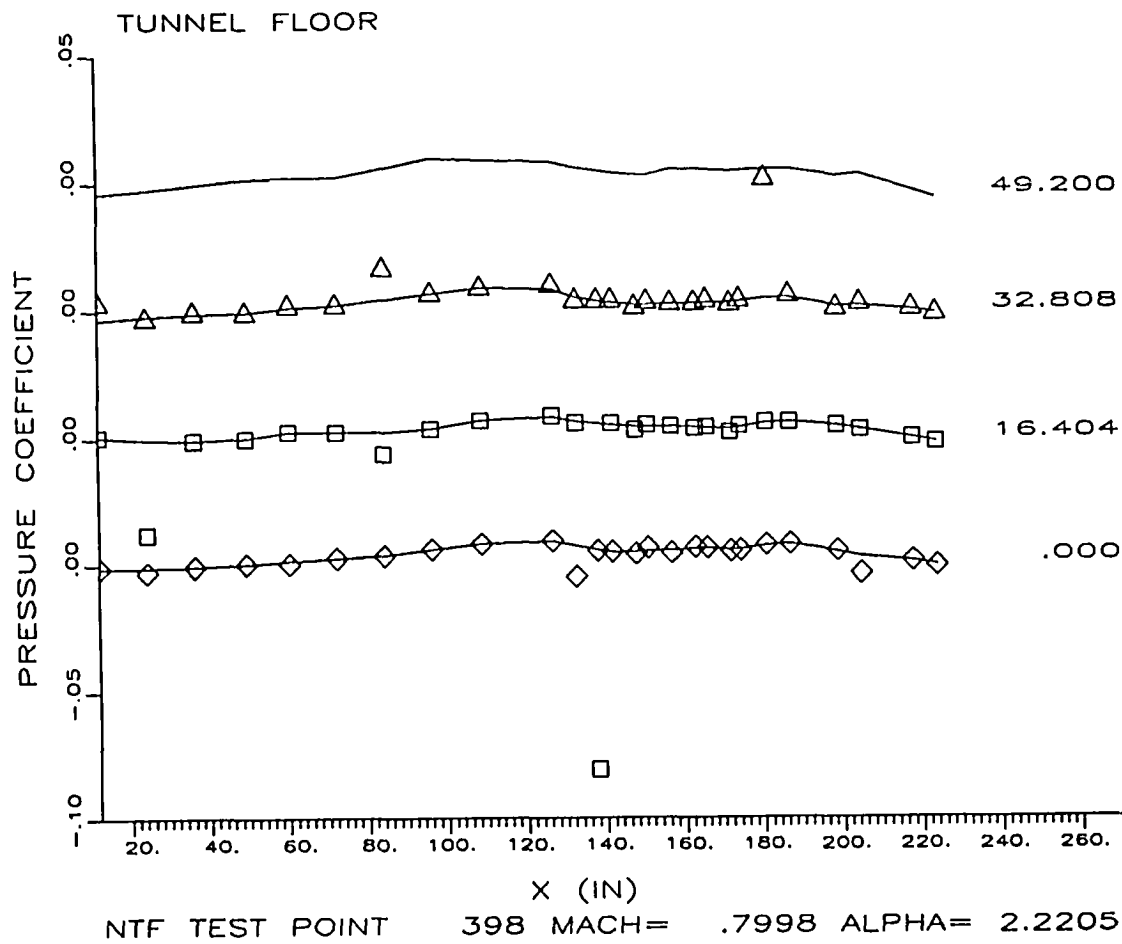


Figure 6 (b). Plot of Experimental (Symbols), Smoothed (Solid Lines), and Interpolated (Solid Lines With No Corresponding Symbols) Lower Wall Pressures.

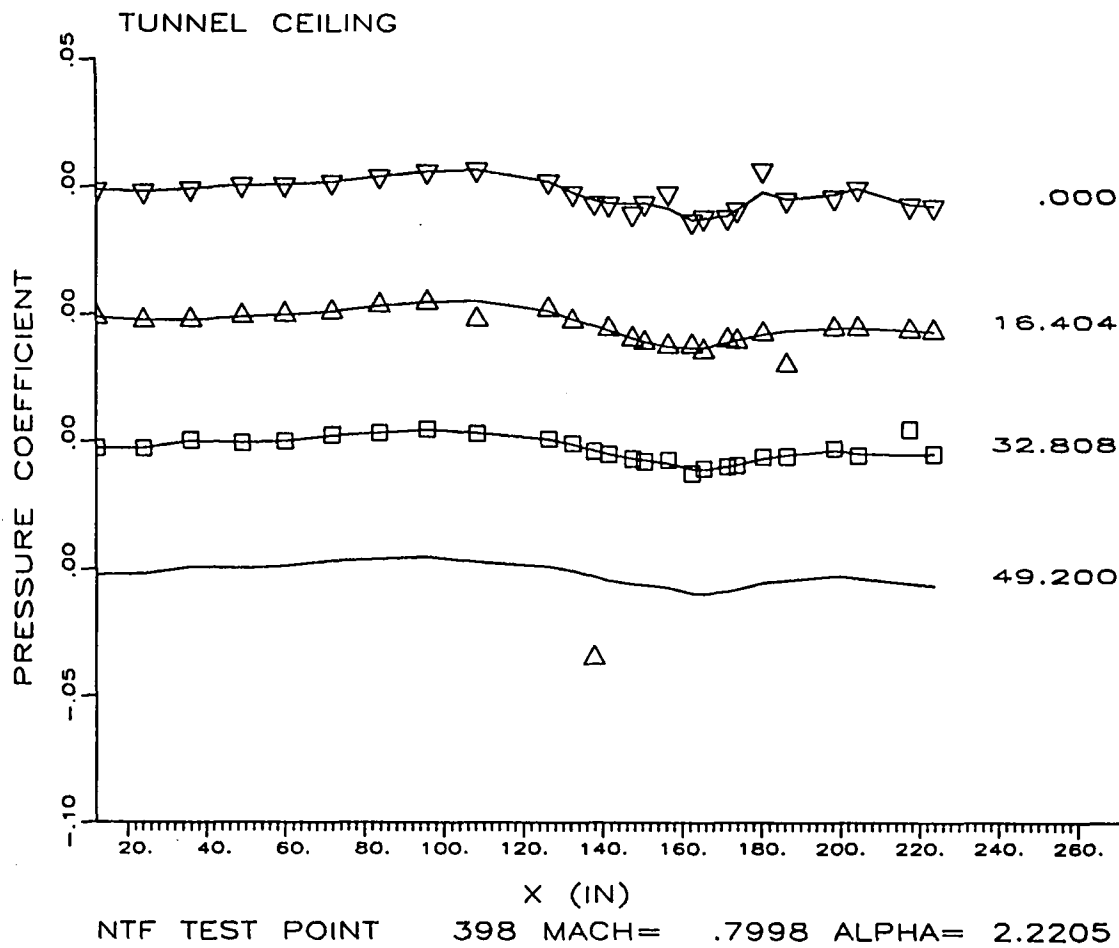


Figure 6 (c). Plot of Experimental (Symbols), Smoothed (Solid Lines), and Interpolated (Solid Lines With No Corresponding Symbols) Upper Wall Pressures.

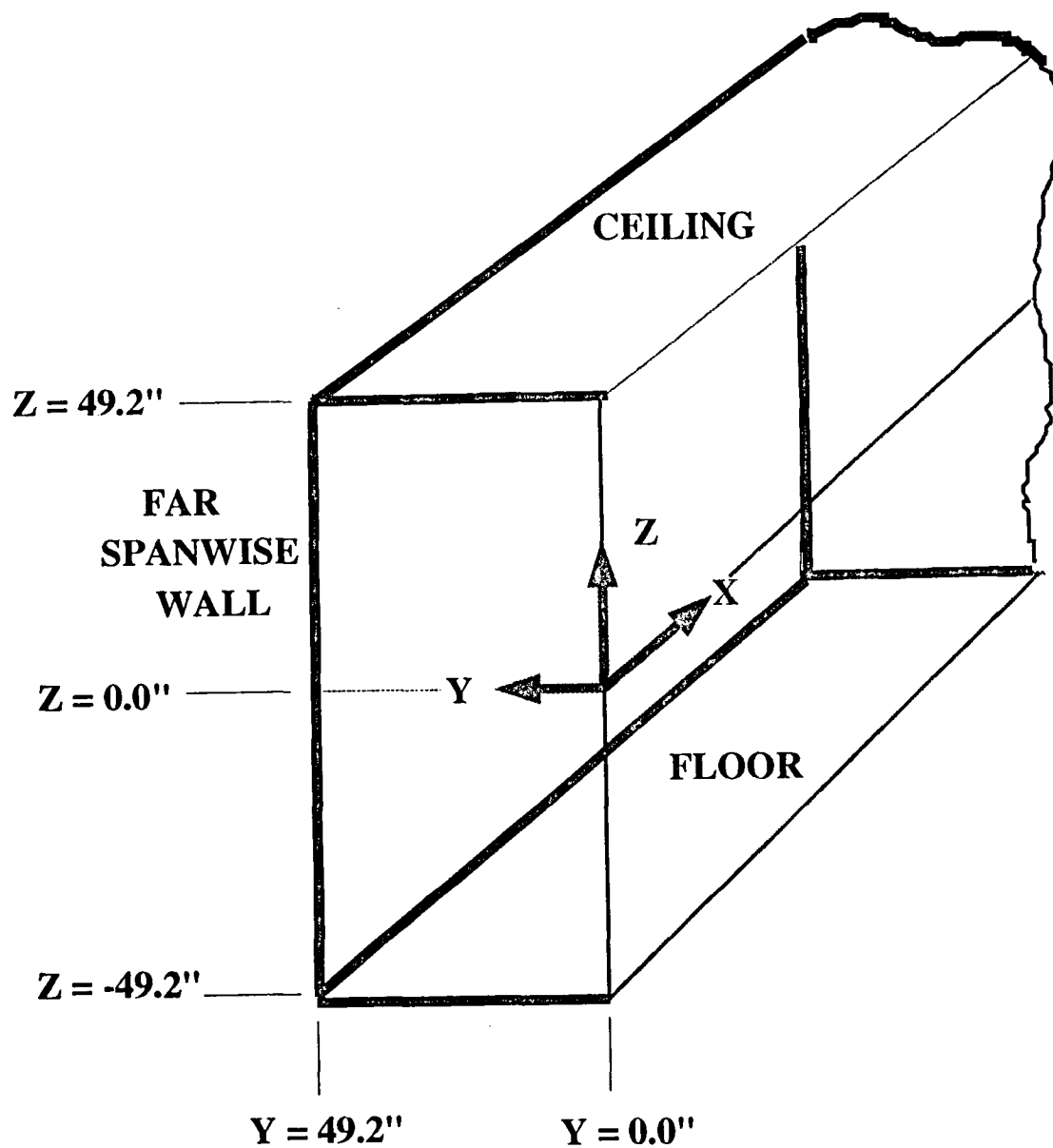


Figure 7. Coordinate System Used in WIACX Codes.

```

*****
***** wiacx code parameters *****
*****pathfinder 1 in ntf*****
***** tunnel and model description-related parameters: *****
      nhs
      2
      nwing      ntail
      1          0
      0          1
      neufsw      neuw      neuuw      ipnlw      ipnuw
      0          0          0          0          0
***** correction-scheme-related parameters: *****
      nsing      mdim      mchw      mcht      mchf
      2          2367      1          1          1
      ca          xinc      erra      xincm0
      1.00000    0.10000    0.00005    0.00050
      c1          c2          errm
      1.20000    0.60000    0.00050
*****

```

Figure 8. Sample "Fort.38" file.

```

#
# QSUB-lt 7200
# QSUB-lm 40mw
# QSUB-me
#
cd /scr/WIACX.demol
cft -M 4000 wiacxt.f
ldr wiacxt.o
sh -c "a.out <pathf.tunnel >pathft.out"
mv fort.24 fort.4
mv fort.21 fort.21t
mv fort.31 fort.31t
mv fort.14 fort.14t
cft -M 4000 wiacxf.f
ldr wiacxf.o
sh -c "a.out <pathf.frecair >pathff.out"
mv fort.21 fort.21f
mv fort.31 fort.31f
mv fort.14 fort.14f
exit

```

Figure 9. Sample Submittal File.

```

test 21 run 18 point 398
experimental conditions:
cl = 0.516640E+00
cd = 0.363300E-01
cm = -0.693140E+00
mach = 0.799790E+00
computed conditions:
delta - alpha (deg): 0.268000E-01
delta-cd due to pressure gradient: 0.386992E-05
delta-cd due to rotation: 0.241657E-03
q-ratio, q-exp/q-corr: 0.100656E+01
corrected cl = 0.520029E+00
corrected cd = 0.368154E-01
corrected cm = -0.697686E+00

```

Figure 10. Sample Output From Bouyancy Correction Post-processor.

```

fort.31t
$inpl
n1t2fa=1,nconf=2,nsing=2,
$end

```

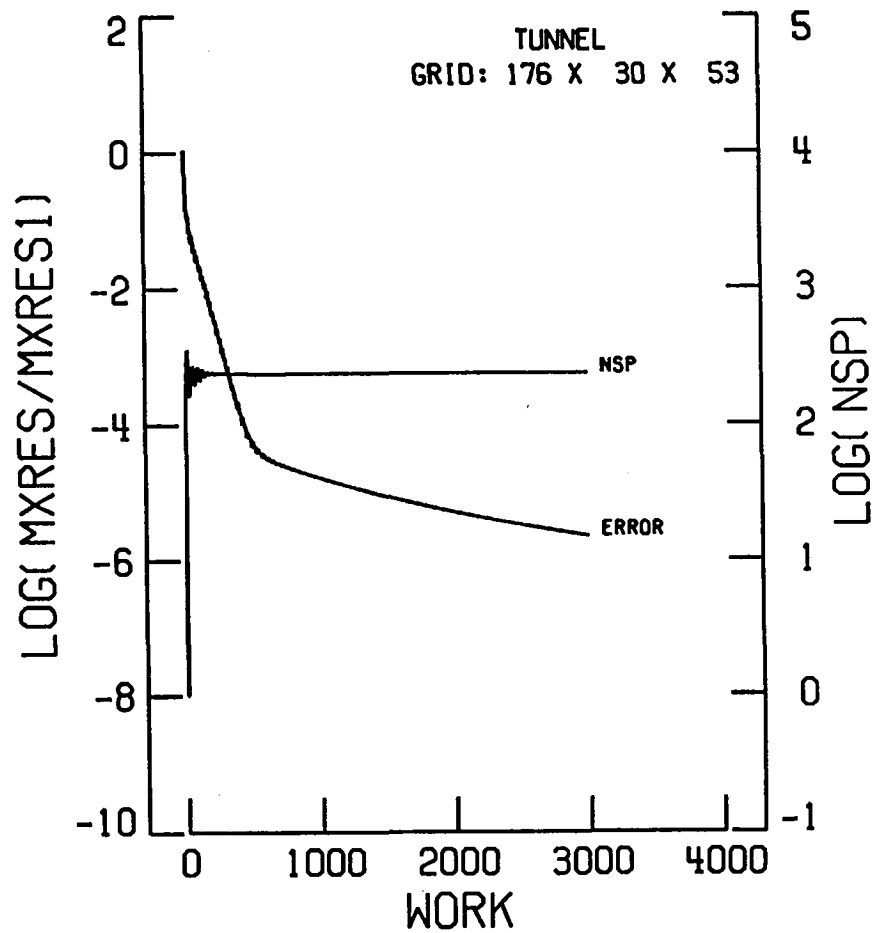
Figure 11. Contents of sample "fort.31t" namelist plot input file.

```

fort.31f
$inpl
n1t2fa=2,nconf=2,nsing=1,
$end

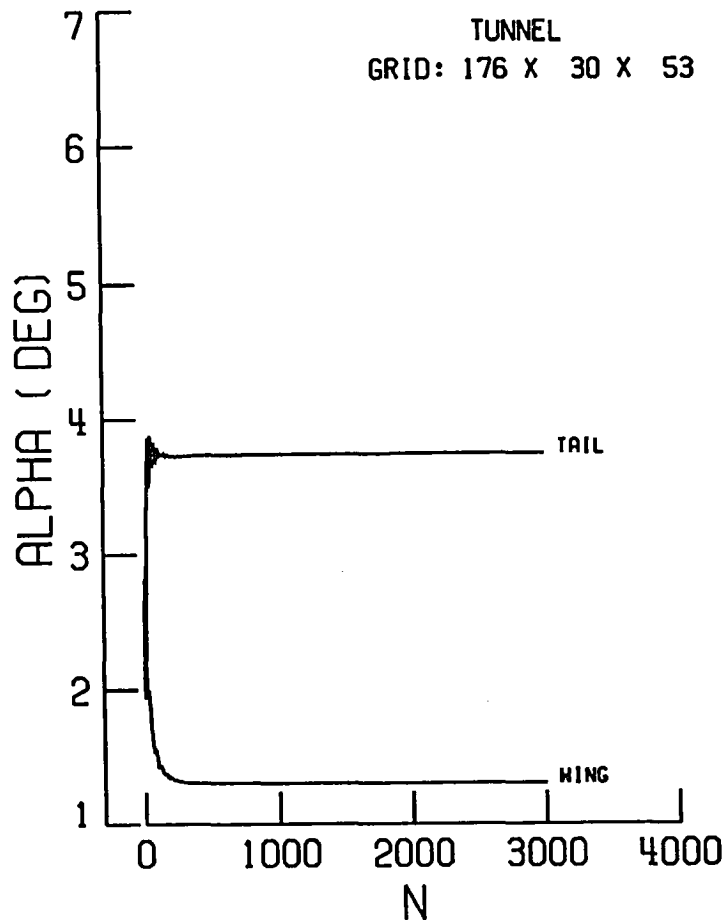
```

Figure 12. Contents of sample "fort.31f" namelist plot input file



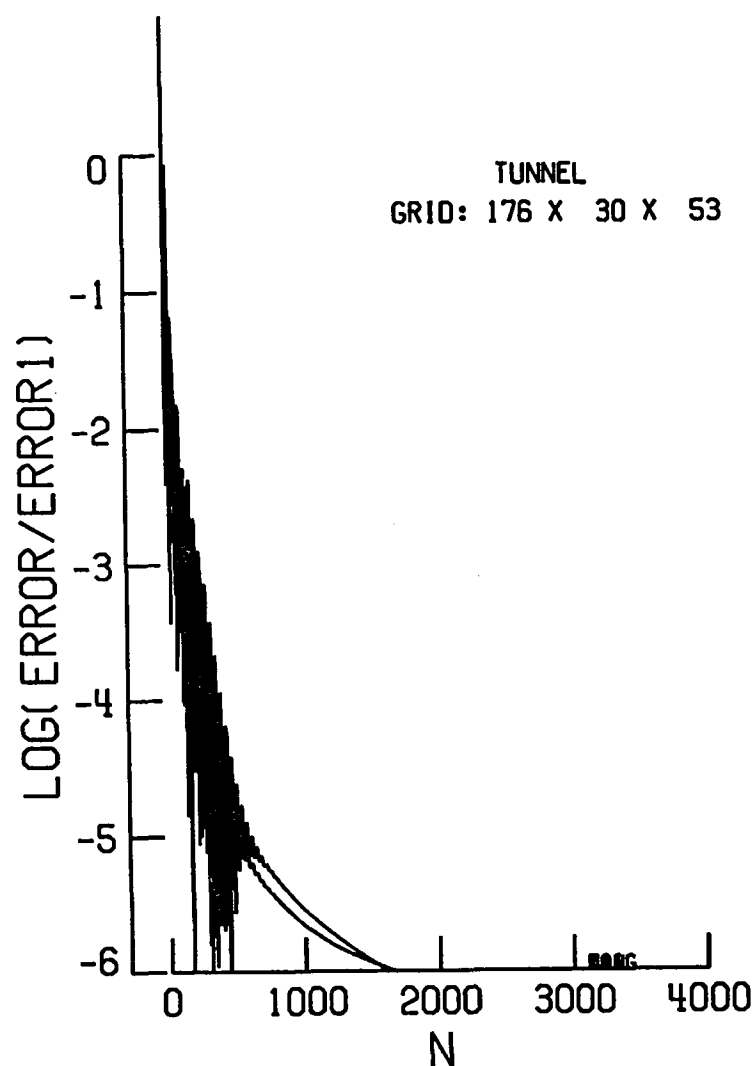
INPUT: AL W=0.2246E+01 AL T=0.2246E+01 XM=.7959
 RES1 0.331E+00 RESN 0.745E-06
 WORK1 0.100E+01 WORKN 0.300E+04 RATE 0.9957
 PATHFINDER 1. TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:39

Figure 13. "In-tunnel" residual and supersonic point history.



PATHFINDER 1. TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:39
 INPUT: AL W=0.2246E+01 AL T=0.2246E+01 XM=.7959
 CALC : AL W=0.1311E+01 AL T=0.3743E+01 XM=.7959

Figure 14. "In-tunnel" wing and tail angle-of-attack history.



PATHFINDER 1. TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:39
 WING ERR 1 -0.715E+00 WING ERR N 0.174E-06
 TAIL ERR 1 -0.185E+01 TAIL ERR N -0.660E-06

Figure 15. "In-tunnel" wing and tail lift error history.

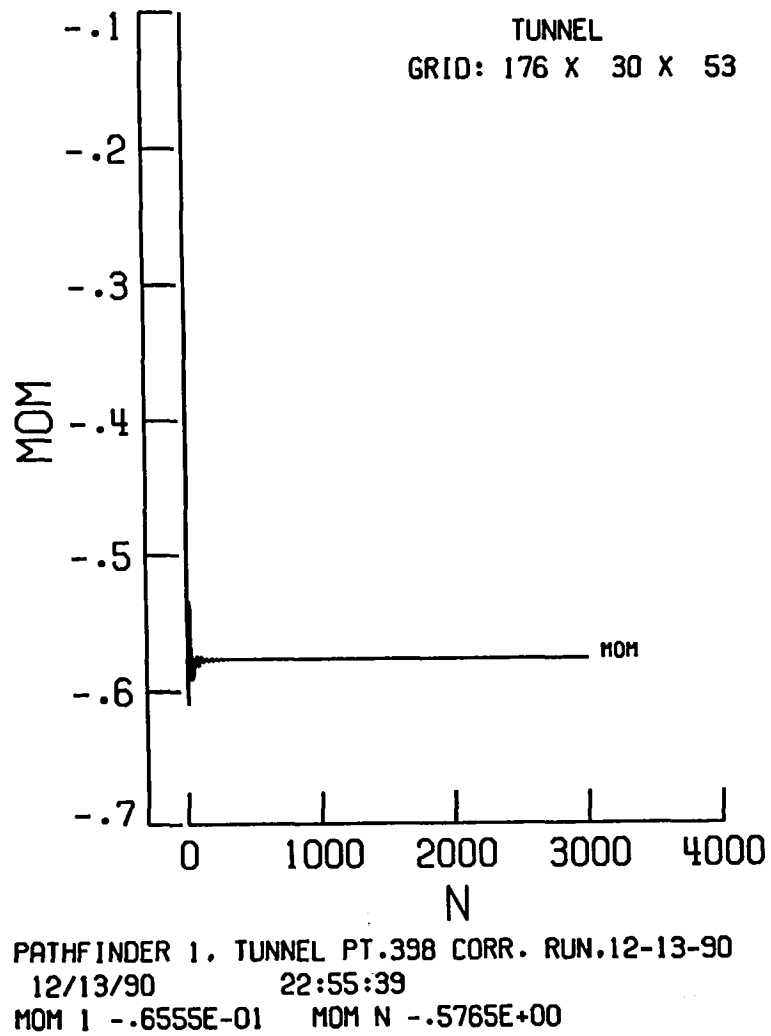
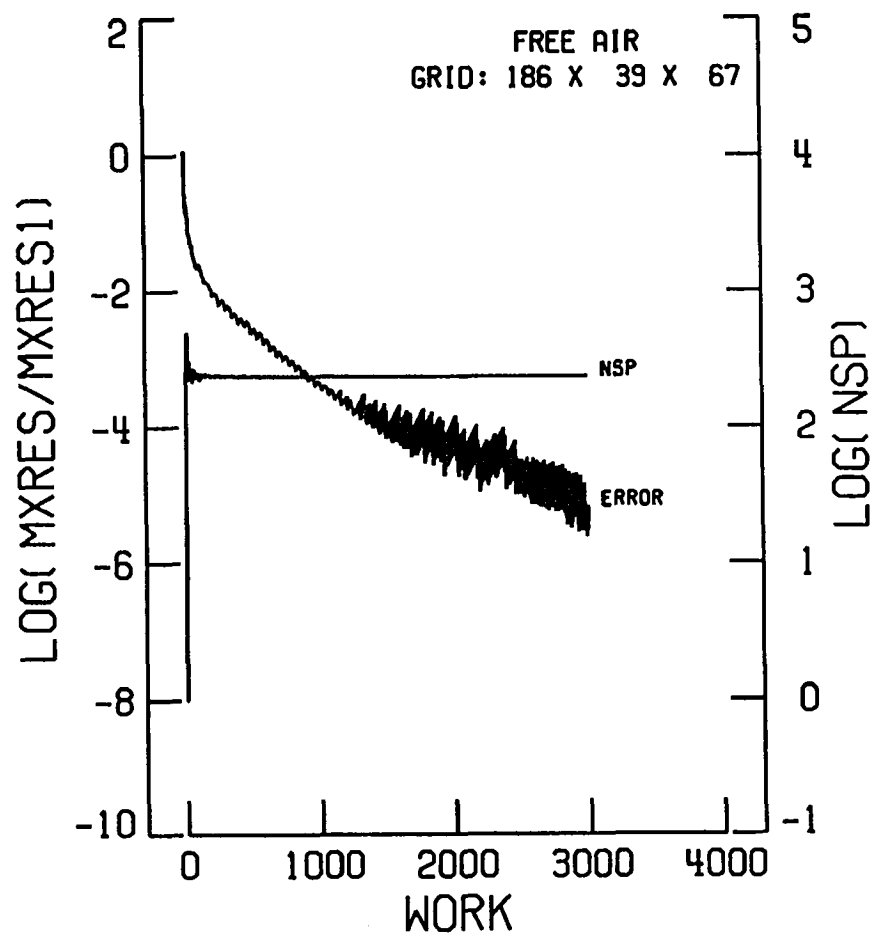
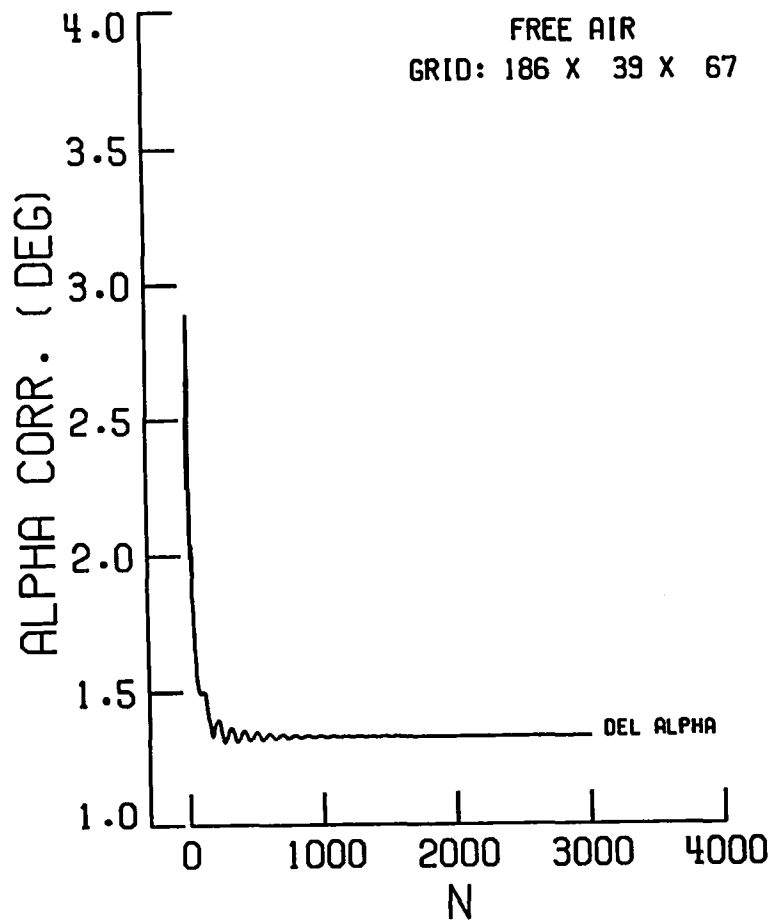


Figure 16. "In-tunnel" configuration pitching moment coefficient history.



ALCOR 1: 0.2246E+01 XM=.7959
 RES1 0.331E+00 RESN -0.221E-05
 WORK1 0.100E+01 WORKN 0.300E+04 RATE 0.9960
 PATHFINDER 1.FREE-AIR PT.398 CORR. RUN. 12-13-90
 12/14/90 0:06:20

Figure 17. Free-air residual and supersonic point history.



PATHFINDER 1.FREE-AIR PT.398 CORR. RUN. 12-13-90

12/14/90 0:06:20

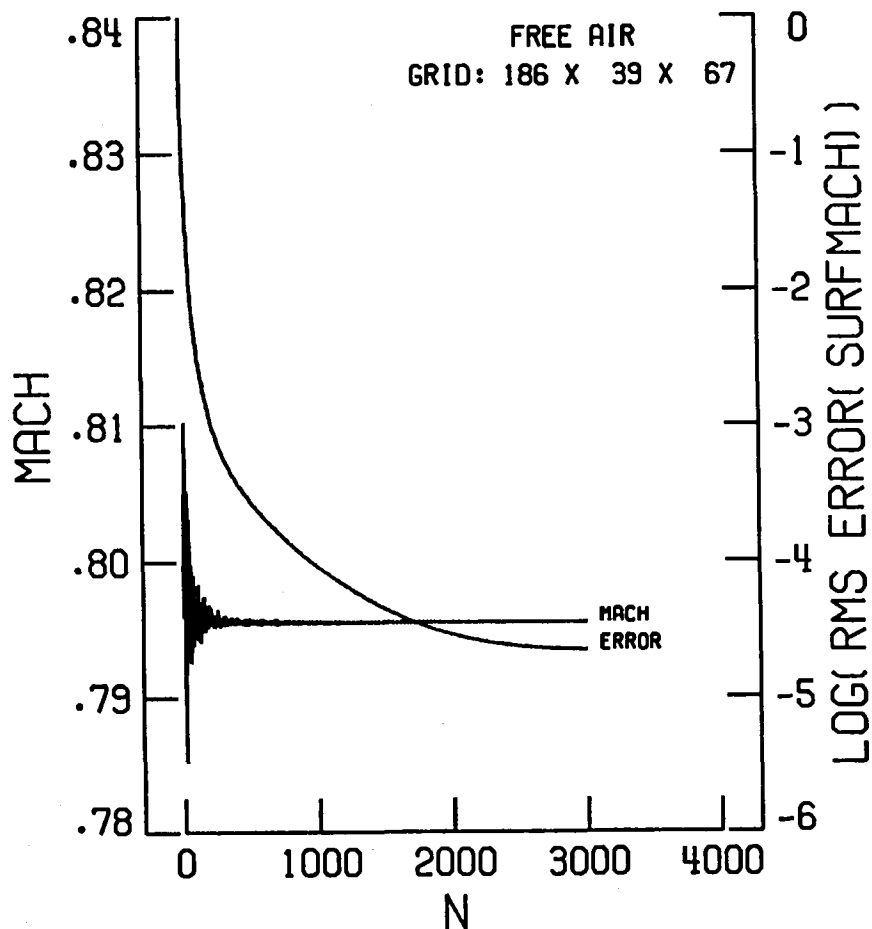
ALCOR 1: 0.2246E+01 XM=-.7959

ALCOR N: 0.1330E+01 XM=-.7955

SINGLE ALPHA CORRECTION

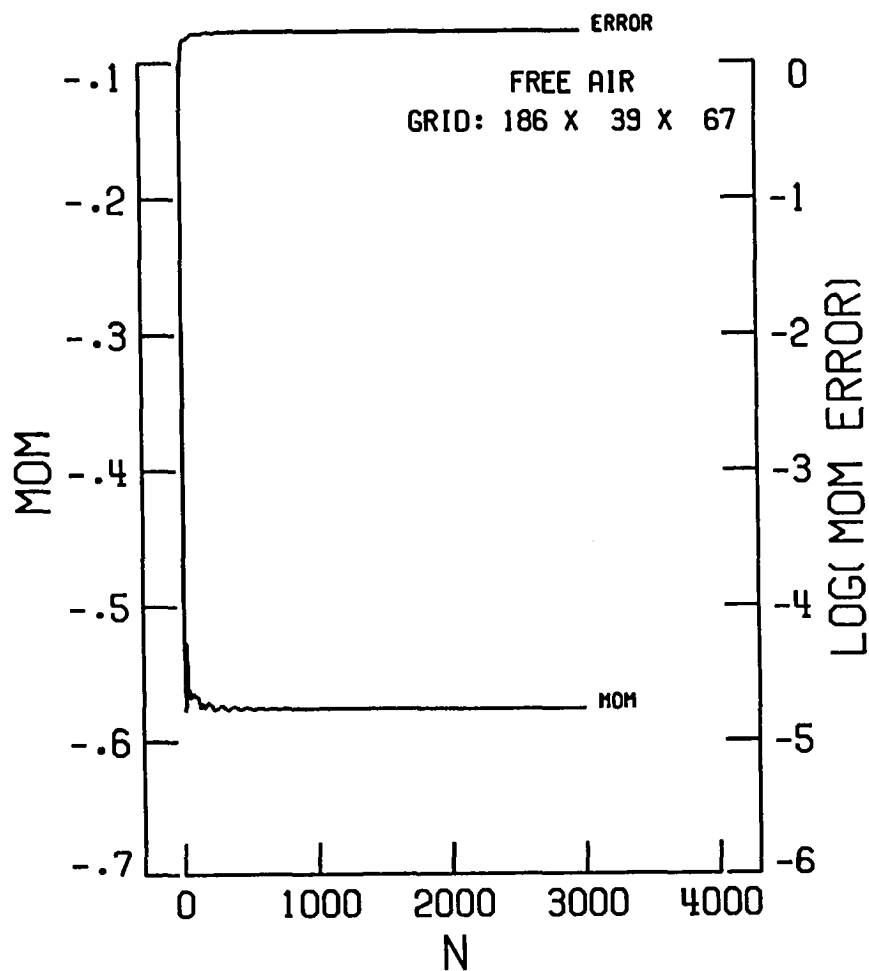
DELTIME= 0.50E+01

Figure 18. Free air delta-alpha history.



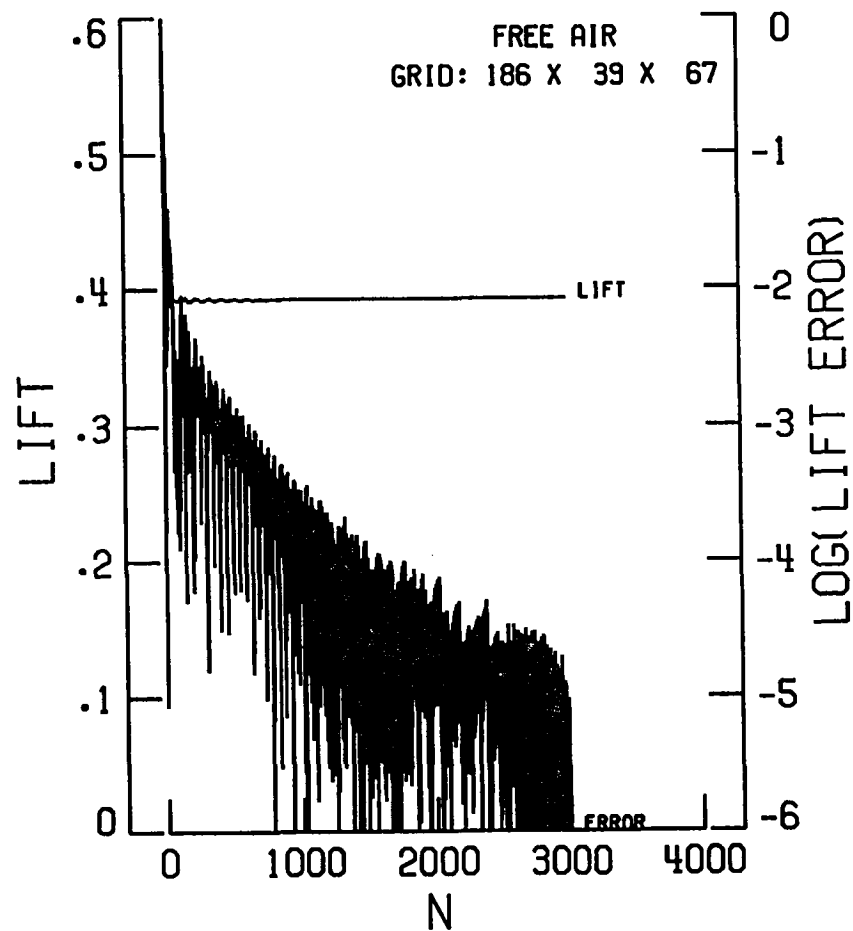
PATHFINDER 1.FREE-AIR PT.398 CORR. RUN. 12-13-90
 12/14/90 0:06:20
 XM 1 0.7959E+00 XM N 0.7955E+00
 RMS ERROR 1 0.147E-01 RMS ERROR N 0.325E-06
 SINGLE ALPHA CORRECTION DELTIME= 0.50E+01

Figure 19. Free air Mach number and RMS error history.



PATHFINDER 1.FREE-AIR PT.398 CORR. RUN. 12-13-90
 12/14/90 0:06:20
 MOM I -.6523E-01 MOM N -.5764E+00
 SINGLE ALPHA CORRECTION DELTIME= 0.50E+01
 MOM ERROR I 0.225E+01 MOM ERROR N 0.380E+01

Figure 20. Free air configuration pitching moment and pitching moment error history.



PATHFINDER 1.FREE-AIR PT.398 CORR. RUN. 12-13-90
 12/14/90 0:06:20
 LIFT 1 0.9271E-01 LIFT N 0.3925E+00
 SINGLE ALPHA CORRECTION DELTIME= 0.50E+01
 LIFT ERROR 1 -0.723E+00 LIFT ERROR N 0.657E-06

Figure 21. Free air configuration lift and lift error history.

```

fort.21t
$nm1 itplt = 3000, nconf = 4,
kactw =0,inexpw=1,spanw=26.54,
kactt=1,spant=9.95,
nyosa(2) = 1, nysa(1,2) = 6,
nfyosa = 2, nfysa(1) = 2,3,
nvalif = 2, nif = 80, 120,
$END
excp398
fort.14t

```

Figure 22. Contents of sample "fort.21t" plot input file.

test	run	point
21	18	398
mexp	mcor	
.79979	.79551	
ns		
6		
eta	ncpu	ncpl
.13100	14	10
upper surface		
x/c	cp	
.02500	-.28300	
.07500	-.49400	
.12500	-.57900	
.20000	-.66300	
.30000	-.74000	
.40000	-.75500	
.45000	-.53300	
.50000	-.46800	
.55000	-.43200	
.60000	-.40000	
.65000	-.35400	
.70000	-.31000	
.80000	-.21700	
.90000	-.07600	
lower surface		
x/c	cp	
.02500	.18200	
.07500	.05500	
.12500	-.06000	
.20000	-.19300	
.30000	-.26200	
.40000	-.28900	
.50000	-.30200	
.60000	-.21000	
.70000	.07200	
.80000	.21200	

Figure 23. First entries in experimental wing pressure file.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:3
 ITERATION 3000 Y/S=-.139 WING
 MACH NO. .796 GAMMA= 1.40
 EXP. ETA(1)= .1310

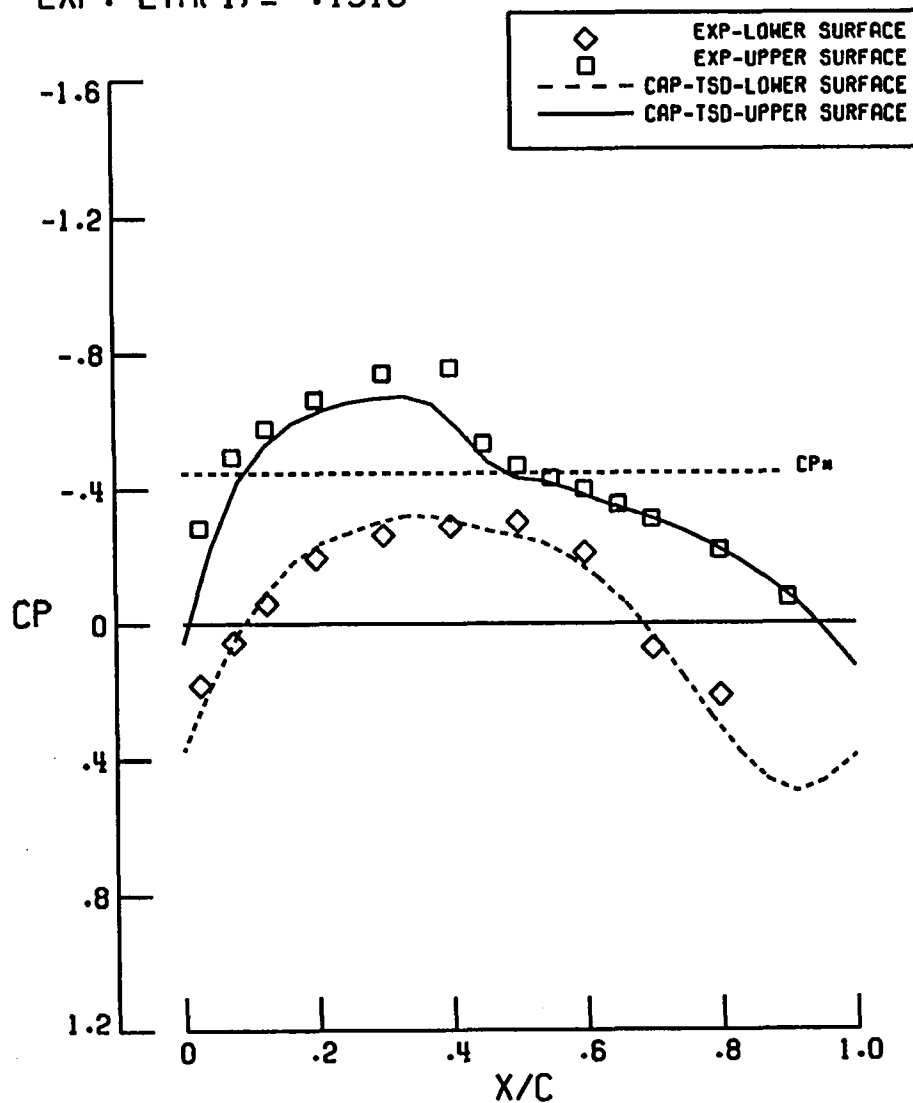


Figure 24. Computed "in-tunnel" wing pressures versus experiment.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:3
 ITERATION 3000 Y/S=.282 WING
 MACH NO. .796 GAMMA= 1.40

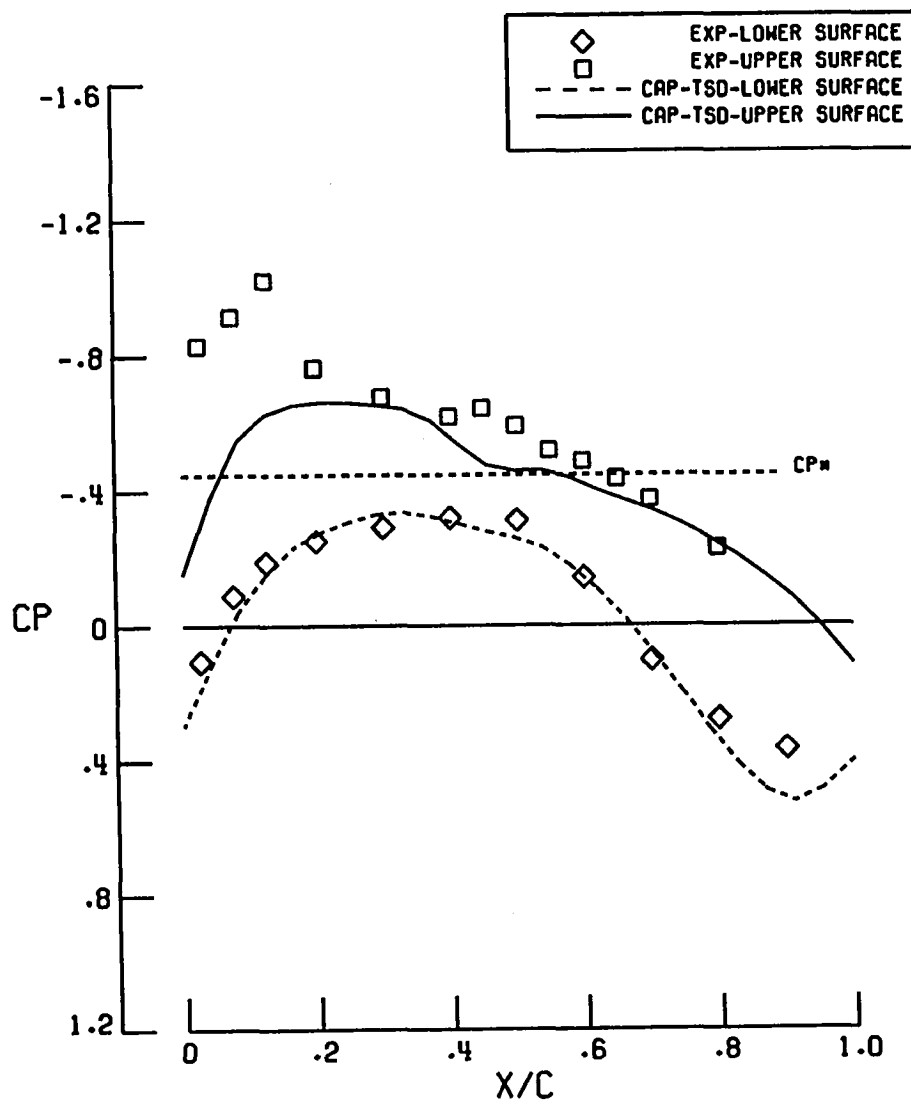


Figure 25. Computed "in-tunnel" wing pressures versus experiment.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:3
 ITERATION 3000 Y/S=.432 WING
 MACH NO. .796 GAMMA= 1.40

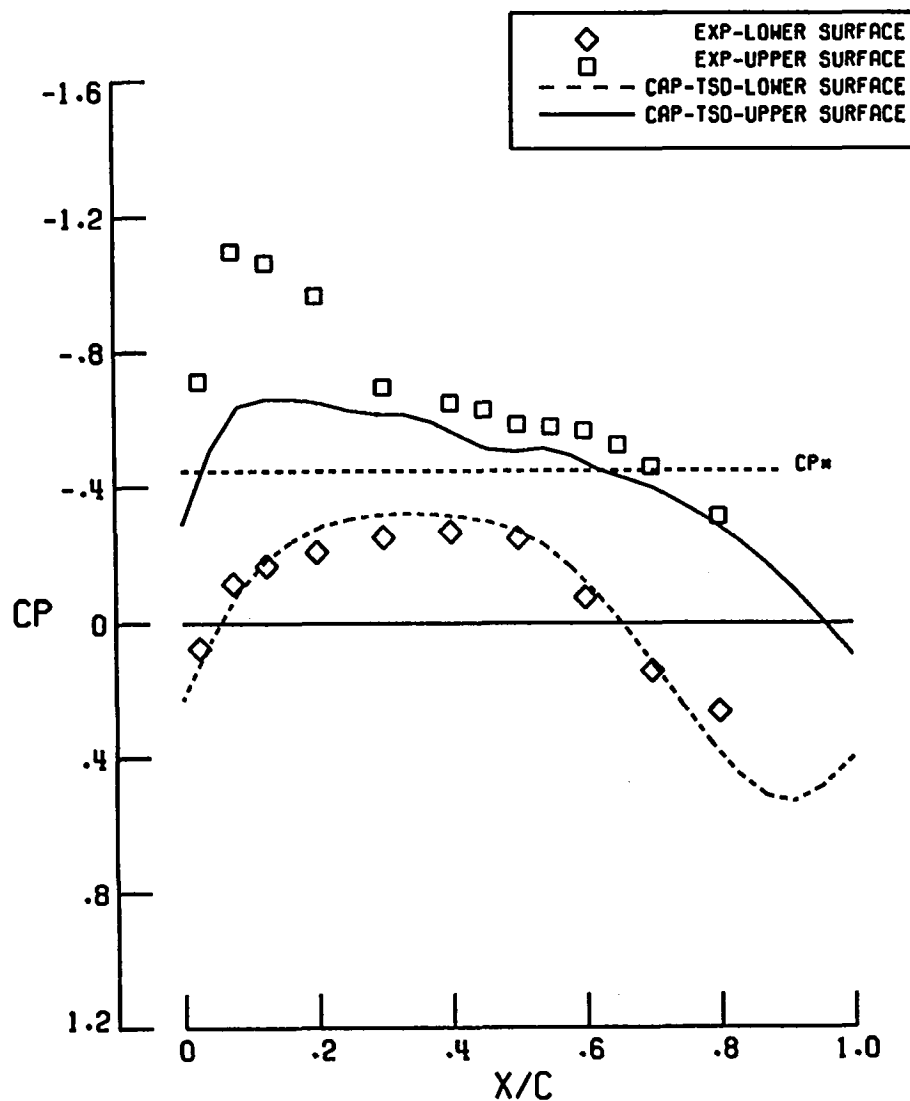


Figure 26. Computed "in-tunnel" wing pressures versus experiment.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:3
 ITERATION 3000 Y/S=.640 WING
 MACH NO. .796 GAMMA= 1.40

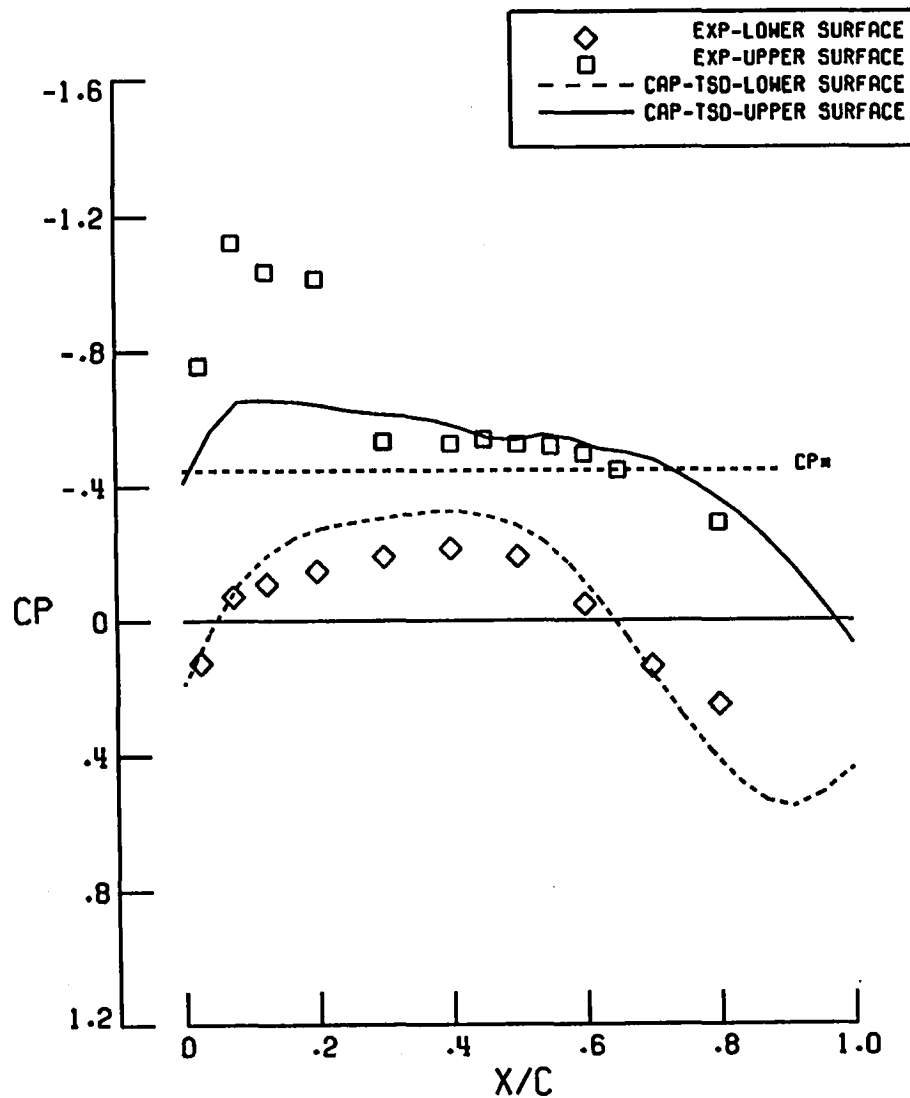


Figure 27. Computed "in-tunnel" wing pressures versus experiment.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:3
 ITERATION 3000 Y/S=.829 WING
 MACH NO. .796 GAMMA= 1.40

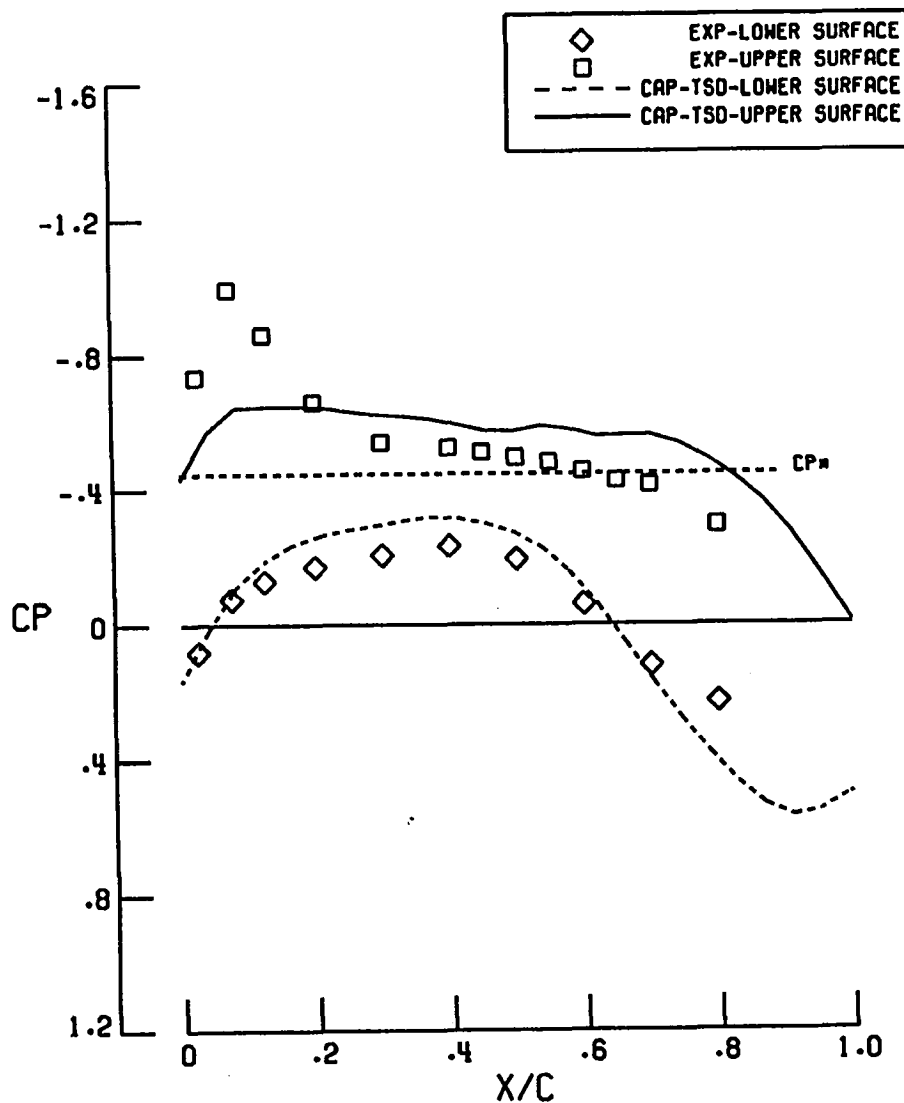


Figure 28. Computed "in-tunnel" wing pressures versus experiment.

PATHFINDER 1. TUNNEL PT.398 CORR. RUN.12-13-90
 12/13/90 22:55:3
 ITERATION 3000 Y/S=.961 WING
 MACH NO. .796 GAMMA= 1.40

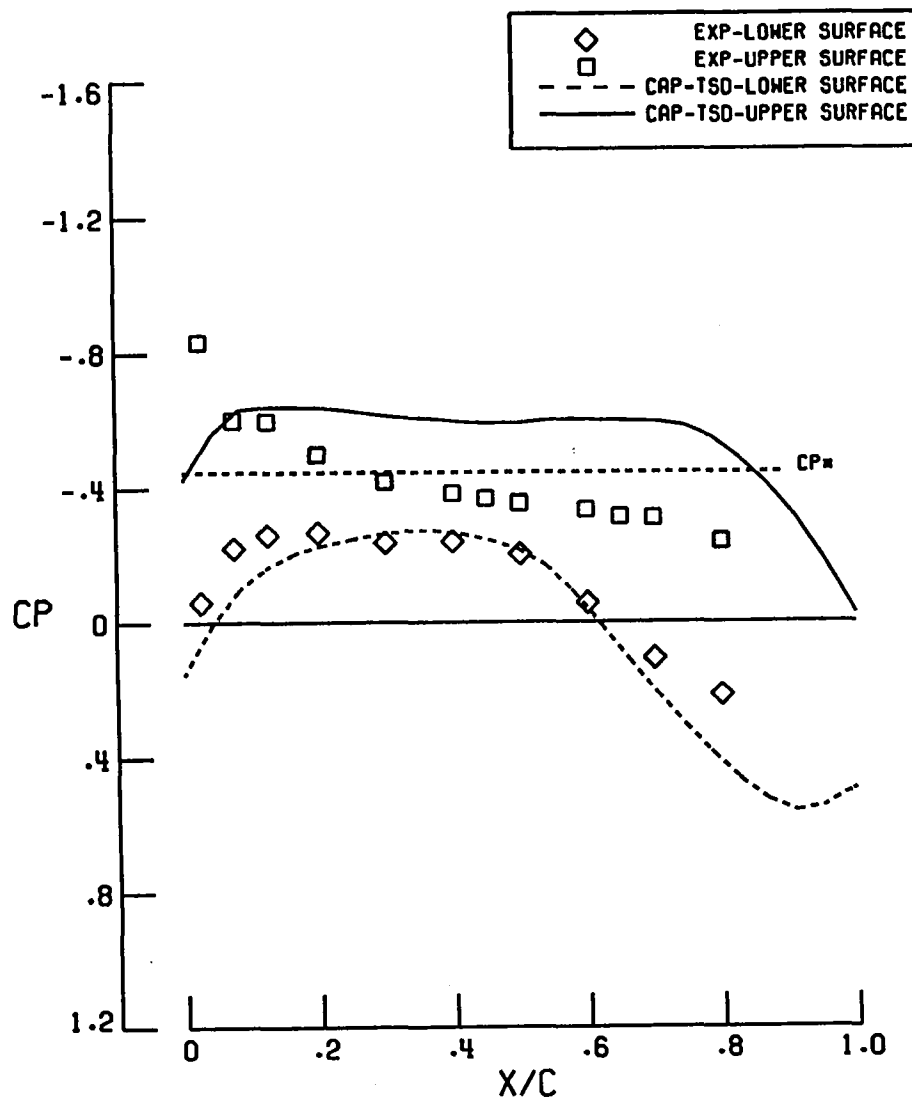


Figure 29. Computed "in-tunnel" wing pressures versus experiment.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
12/13/90 22:55:3
ITERATION 3000 Y-STAT=6 TAIL
MACH NO. .796 GAMMA= 1.40

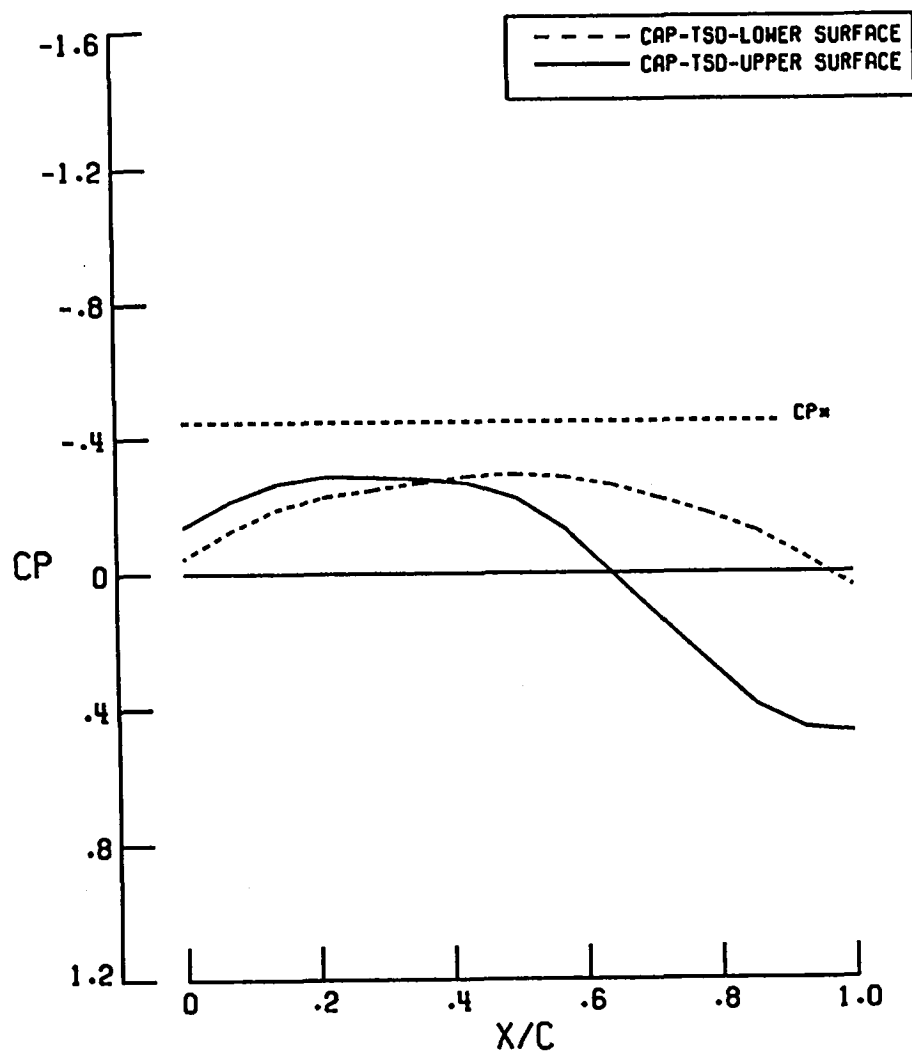


Figure 30. Computed "in-tunnel" tail pressures.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
12/13/90 22:55:3
ITERATION 3000 Y-STAT=2 FUSELAGE
MACH NO. .796 GAMMA= 1.40

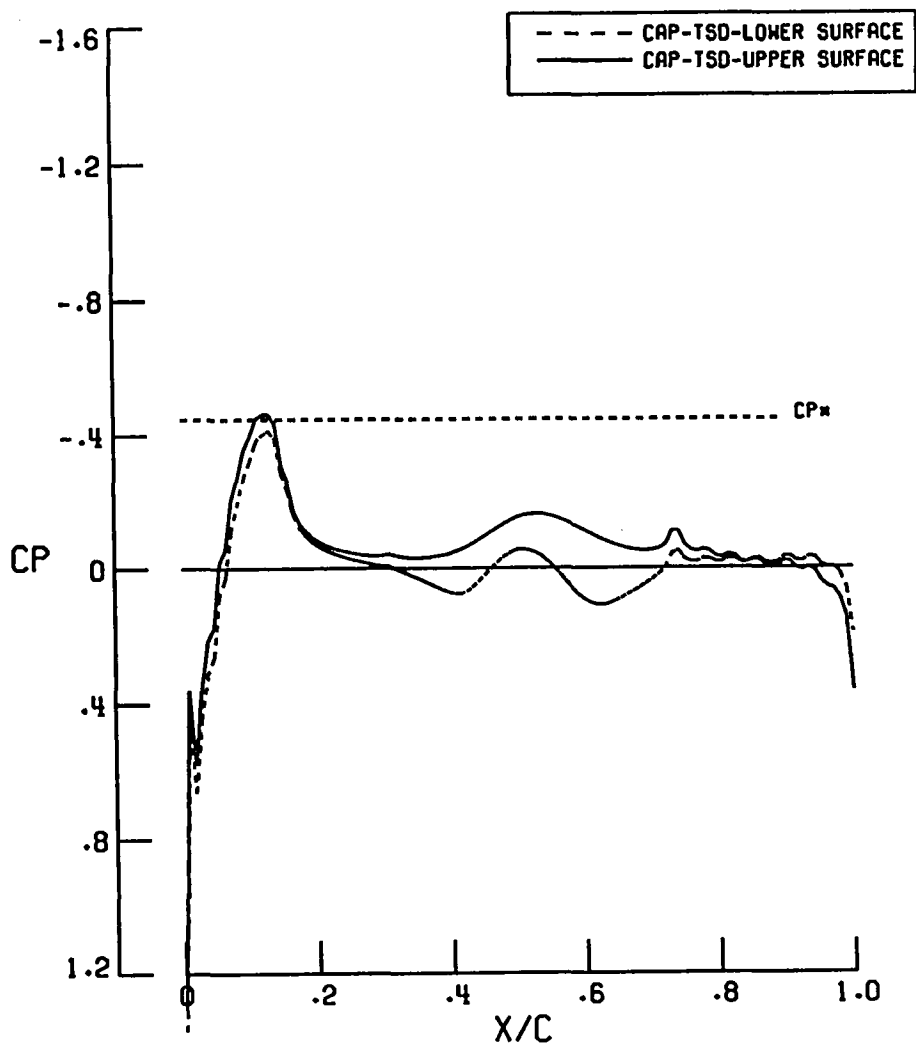


Figure 31. Computed "in-tunnel" longitudinal fuselage pressures.

PATHFINDER 1, TUNNEL PT.398 CORR. RUN.12-13-90
12/13/90 22:55:3
ITERATION 3000 Y-STAT=3 FUSELAGE
MACH NO. .796 GAMMA= 1.40

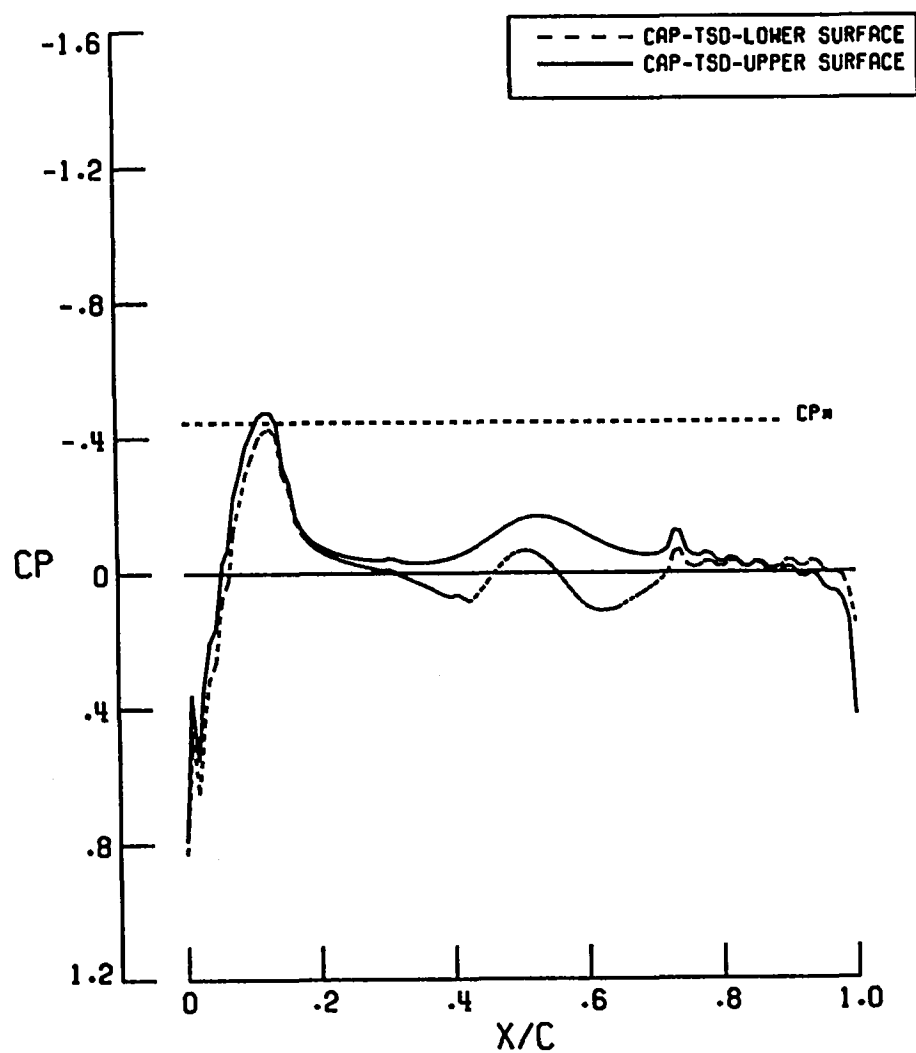


Figure 32. Computed "in-tunnel" longitudinal fuselage pressures.

PATHFINDER 1. TUNNEL PT.398 CORR. RUN.12-13-90
12/13/90 22:55:3
ITERATION 3000 S-STAT=80 FUSELAGE
MACH NO. .796 GAMMA= 1.40

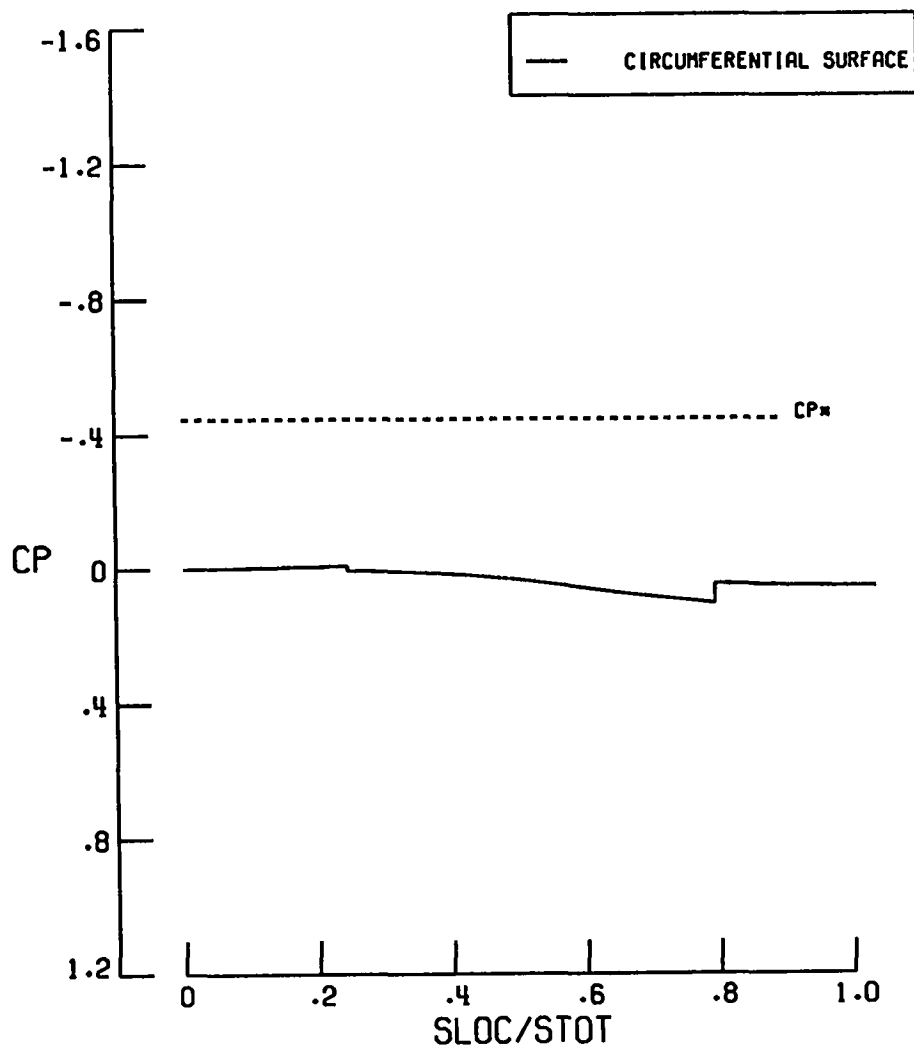


Figure 33. Computed "in-tunnel" circumferential fuselage pressures.

PATHFINDER 1. TUNNEL PT.398 CORR. RUN.12-13-90
12/13/90 22:55:3
ITERATION 3000 S-STAT=120 FUSELAGE
MACH NO. .796 GAMMA= 1.40

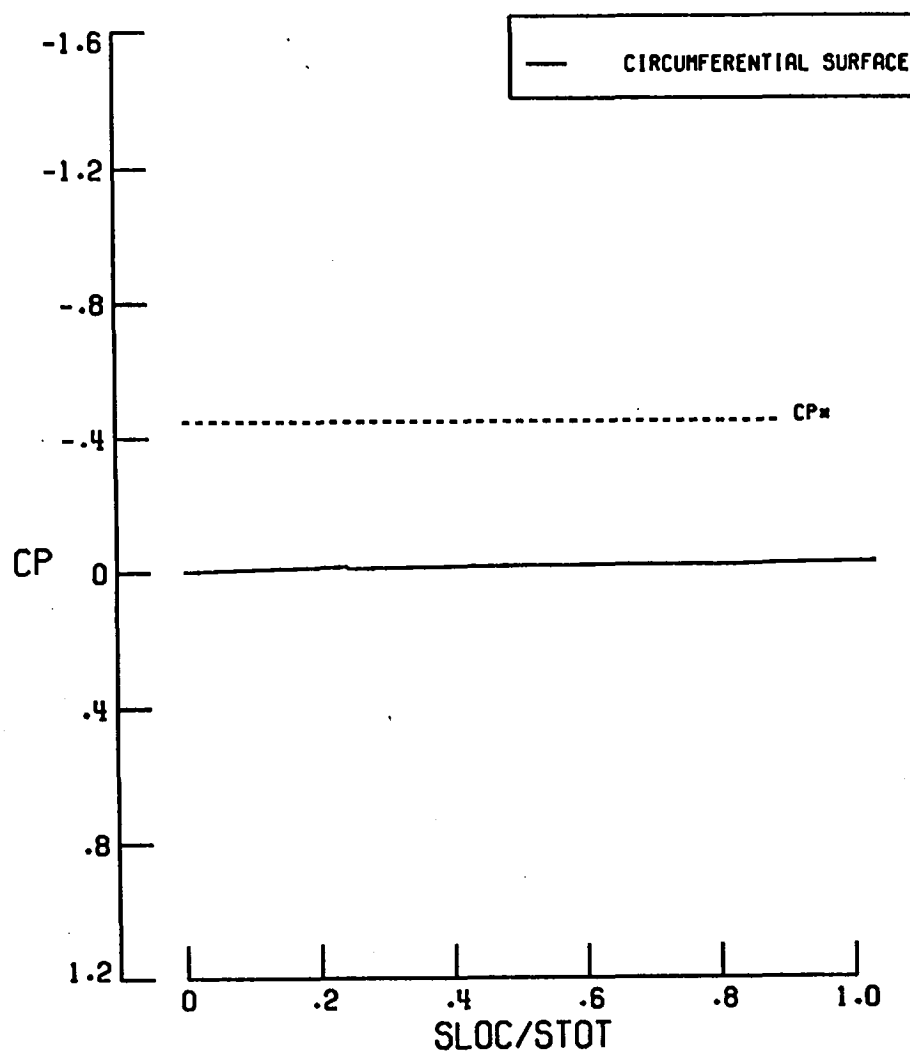


Figure 34. Computed "in-tunnel" circumferential fuselage pressures.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1992		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE User Guide for WIACX: A Transonic Wind-Tunnel Wall Interference Assessment and Correction Procedure for the NTF			5. FUNDING NUMBERS WU 505-59-53-01	
6. AUTHOR(S) Javier A. Garriz Kara J. Haigler				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23665-5225			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA TM-104168	
11. SUPPLEMENTARY NOTES Garriz: Vigyan, Inc., Hampton, VA 23666 Haigler: NASA Langley Research Center, Hampton, VA 23665-5225				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 02			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) WIACX is a three-dimensional transonic wind-tunnel interference assessment and correction (WIAC) procedure developed specifically for use in the National Transonic Facility (NTF) at NASA Langley Research Center. This report is a user manual for the codes comprising the correction procedure. It also includes listings of sample procedures and input files for running a sample case and plotting the results.				
14. SUBJECT TERMS Wall Interference Transonic Flow Wind Tunnel Correction			15. NUMBER OF PAGES 66	
			16. PRICE CODE A04	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	

